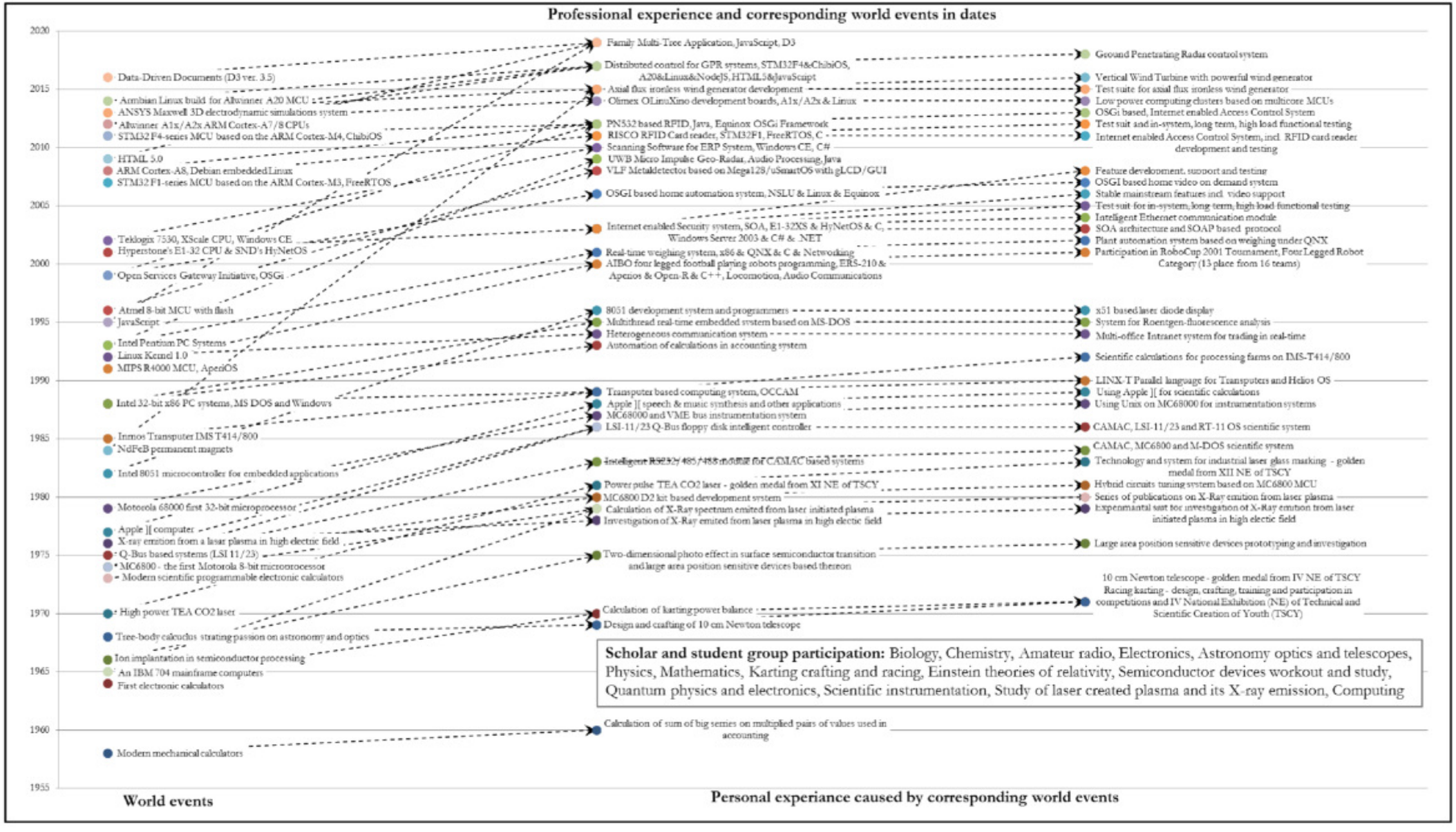
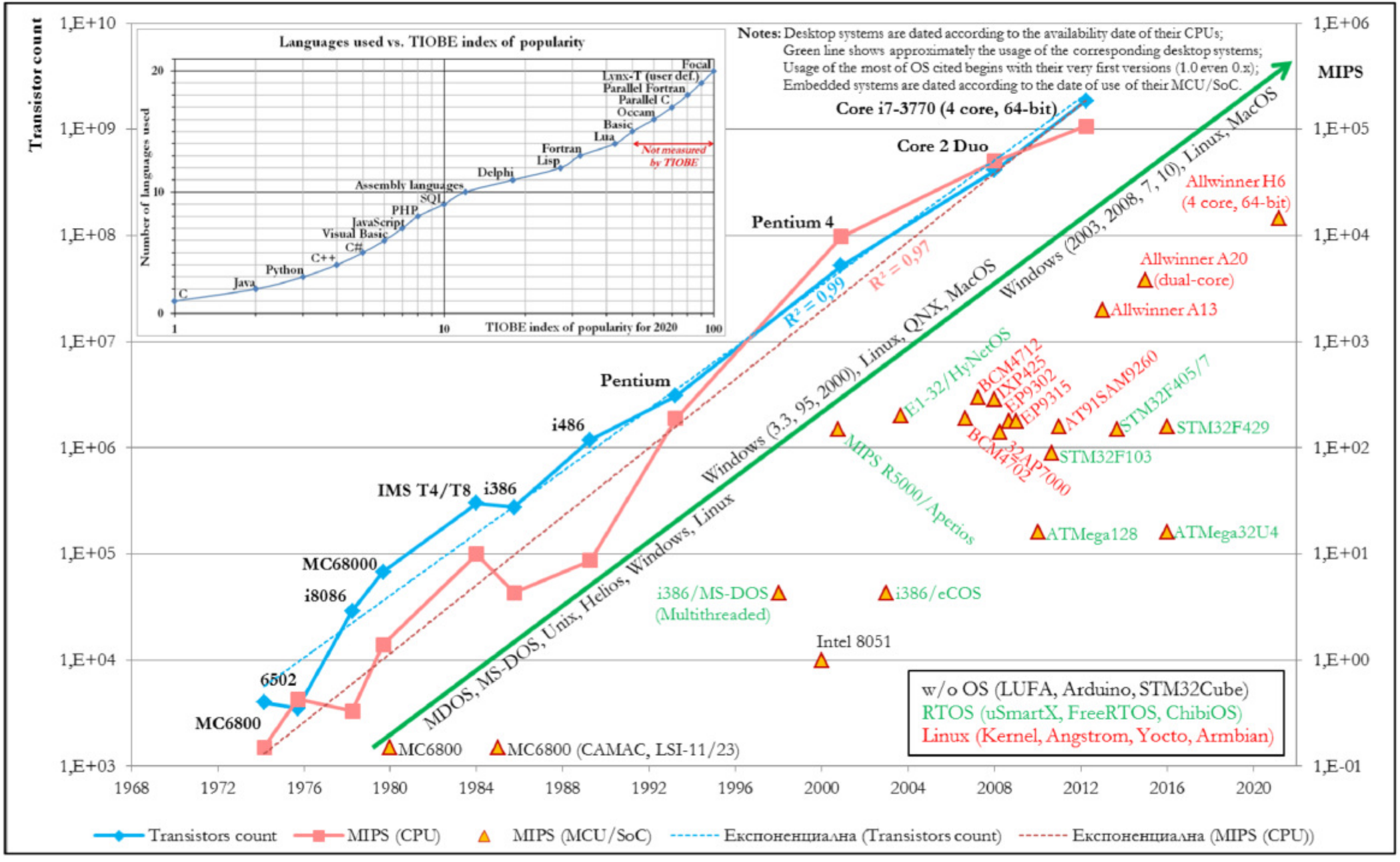
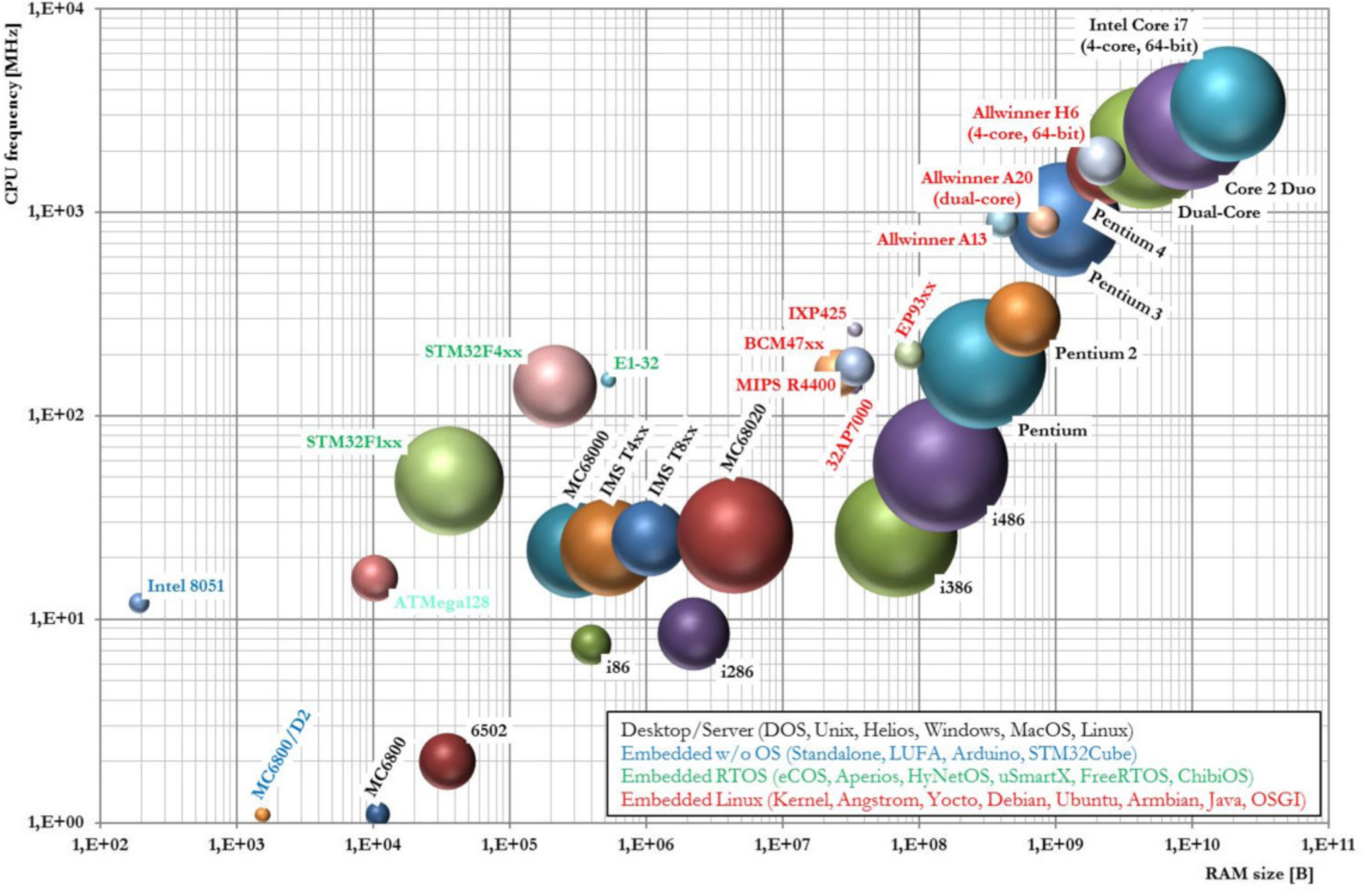
Sources of Christo Radev's professional experience



CPU, MCU, SoC, OS and programming languages used to gain professional experience in the time



CPU, MCU, SoC and system performance used to gain professional experience (CPU frequency vs. RAM size)



Devices, MCUs and SoCs used to gain professional experience in embedded systems

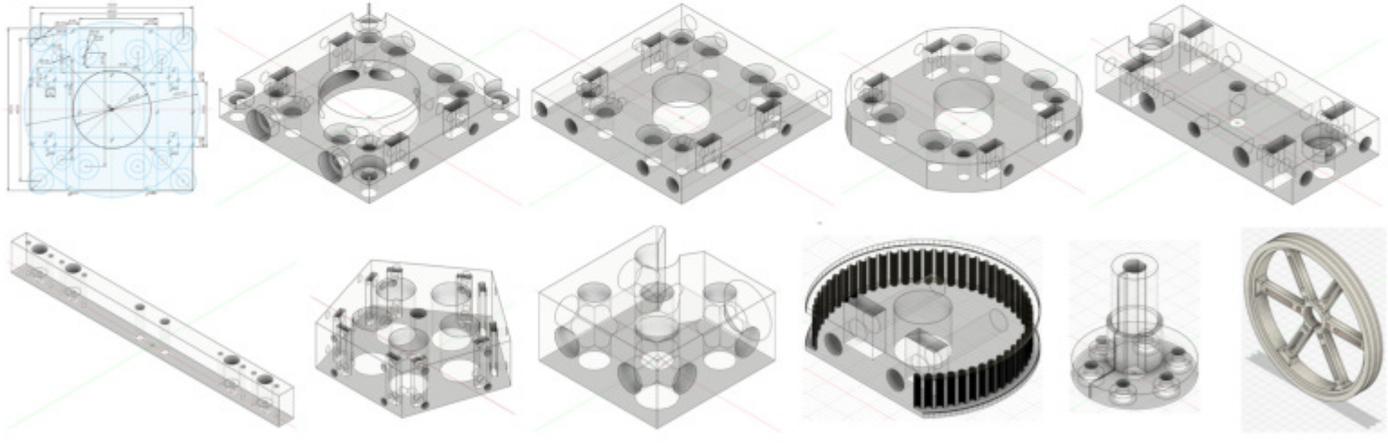
Date	Device	SoC/MCU	Core	MHz	RAM [MB]	FLASH [MB]	100MB Eth	1GB Eth	Wireless	HD	USB 2.0	USB 3.0	Other
1.12.2010	Cinterion TC65 Terminal	TC65 Module (Java IMP-NG profile)							GSM 850/900/1800/1900				SIM, 1xRS232, 2xADC, SPI, I2C, GPIO, Audio Out, Antenna
9.1.2009	Grandstream HandyTone 488	ATA & VoIP					1+1						1xFXS, 1xFXO / PSTN Pass-through
8.10.2015	Linksys WRT1900ACS	Marvell Armada 385 88F6820	2 x ARM v7	1 600	512	128		1+4	802.11a/b/g/n/ac	eSATA	1	1	1xUART, JTAG
1.1.2013	D-Link DIR-860L	Broadcom BCM47081A0	2 x Cortex-A9	880	128	128		1+4	802.11a/b/g/n/ac			1	1xUART
1.12.2005	Linksys WRV54G	Intel IXP425	Intel XScale	266	32	8	1+4		802.11b/g				1xUART, miniPCI, JTAG
1.12.2005	Linksys WRT54GS v.1.0	Broadcom BCM4712	MIPS32	200	32	8	1+4		802.11b/g				1xUART, JTAG
1.3.2005	Netgear WGT634U	Broadcom BCM5365	MIPS32	200	32	8	1+4		802.11b/g		1		1xUART
1.1.2005	Linksys NSLU2	Intel IXP420	Intel XScale	133	32	8	1				2		2xUART, JTAG
1.12.2004	Linksys WRT54G v.1.0	Broadcom BCM4702	MIPS32	125	16	4	1+4		802.11b/g				1xUART, JTAG
1.4.2015	Olimex A20-OLinuXino-LIME2-e8G	Allwinner A20	2 x Cortex-A7	1 000	1GB	8GB		1		SATA 2	2+1-otg		eMMC, uSD, HDMI, Audio I/O, 2xUEXT, GPIO
1.9.2014	Olimex A20-OLinuXino-LIME2	Allwinner A20	2 x Cortex-A7	1 000	1GB	8GB		1		SATA 2	2+1-otg		uSD, HDMI, Audio I/O, 2xUEXT, GPIO
1.4.2014	Olimex A20-SOM-4GB	Allwinner A20	2 x Cortex-A7	1 000	1GB	4GB		1		SATA 2	2+1-otg		uSD, HDMI, LCD, Audio I/O, GPIO
1.4.2014	Olimex A20-SOM-EVB	Allwinner A20	2 x Cortex-A7	1 000	1GB	4GB		1		SATA 2	2+1-otg		uSD, Camera, HDMI, VGA, Audio I/O, 2xUEXT, GPIO
1.1.2014	Olimex A13-OLinuXino-WiFi	Allwinner A13	Cortex-A8	1 000	512	8GB			802.11b/g		3+1-otg		uSD, VGA, Audio I/O, 2xUEXT, GPIO
1.1.2014	Olimex A13-SOM-512 + A13-SOM-WIFI-4	Allwinner A13	Cortex-A8	1 000	512	4GB			802.11b/g		1-otg		uSD, UART, GPIO
1.9.2013	Olimex A10-OLinuXino-LIME	Allwinner A10	Cortex-A8	1 000	512	8GB	1			SATA 2	2+1-otg		uSD, HDMI, LCD, GPIO
1.1.2009	TI BeagleBoard	TI OMAP3530	Cortex-A8	720	256	256					1+1-otg		DVI-D, S-Video, Audio I/O, SD/MMC/SDIO, RS232, GPIO, JTAG
2.4.2007	Gumstix Verdex XL6P	Marvell PXA270	Intel Xscale	600	128	32	1				2		LCD, uSD, 3xUART, AC97, SPI, I2C, GPIO, JTAG
1.9.2013	Olimex iMX233-OLinuXino-Maxi	iMX233	ARM926J	454	64		1				2		uSD, TV Video Out, Audio I/O, UEXT, GPIO
1.10.2014	Olimex RT5350F-OLinuXino	Ralink RT5350 F	MIPS24KEc	360	32	8	5		802.11n		1		5-port ethernet switch, 2xUART, SPI, I2C, I2S,, GPIO, JTAG, Antena
1.10.2014	RT5350F-OLinuXino-EVB rev. A	Olimex RT5350F-OLinuXino	MIPS24KEc	360	32	8	2		802.11n		1		5-port ethernet switch, UART, 2xRelay, UEXT, GPIO
1.9.2009	Glomation GESBC-9302E	CS EP9302	ARM920T	200	64	16	1				2		2xUART, RS232, ADC, DAC, SPI, GPIO, JTAG
24.2.2006	Cirrus Logic EDB9315A	Cirrus Logic EP9315	ARM920T	200	64	16	1			IDE	2		LCD, Audio In/Out, VGA, 3xUART, GPIO, JTAG
1.1.2009	Olimex SAM9-L9260	Atmel AT91SAM9260	ARM926EJ-S	180	64	512	1				1+1-dev		SD/MMC, RS232, UEXT, GPIO,JTAG
17.4.2007	AVR32 NGW100	Atmel 32AP7000	AVR-32	140	32	8	2				1-dev		SD/MMC, RS232, ATTiny24, GPIO,JTAG
1.1.2014	Olimex A13-LCD7"-TS												7" LCD, TS, Cable
1.4.2016	Olimex MOD-WIFI-ESP8266-DEV	EPS8266EX	L106	80	160KB	2	2		802.11 b/g/n		1		SPI, I2C, I2S, ADC, GPIO
1.4.2016	Olimex MOD-WIFI-ESP8266-DEV	MOD-WIFI-ESP8266-DEV	L106	80	160KB	2	2		802.11 b/g/n		1		UART, SPI, I2C, I2S, ADC, GPIO, UEXT, Relay
1.4.2011	STM32W RF Control Kit	STMicroelectronics STM32WL	Cortex M4+M0	48	64KB	256KB			802.15.4				3xUART, 2xSPI, 3xI2C, ADC, DAC, GPIO
1.4.2011	Atmel AVR Raven Kit	ATmega3290P + ATmega1284P + AT86RF23	1+1 AVR-8	20	2/16KB	32/128KB			802.15.4		1		LCD, GPIO
1.4.2011	Atmel ATmega128RFA1-EK1	Atmel ATmega128RFA1	AVR-8	16	16KB	128KB			802.15.4				2xUART, 10-bit ADC, SPI, TWI, GPIO, JTAG, Antenna
1.10.2016	Olimex STM32-E407	STMicroelectronics STM32F407ZGT6	Cortex M4	168	196KB	1	1				1+1-otg		uSD, 6xUART, 3xSPI, 3xI2C, 2xCAN, 3xADC, 2xDAC, UEXT, GPIO, JTAG
1.10.2016	Olimex STM32-H405	STMicroelectronics STM32F405RG	Cortex M4	168	196KB	1					1+1-otg		6xUART, 3xSPI, 3xI2C, 2xCAN, 3xADC, 2xDAC, SDIO, GPIO, JTAG
1.8.2013	NXP mbed	NXP LPC1768	Cortex M3	100	64KB	512KB	1				1-otg		4xUART, CAN 2.0, 3xSPI/SSP, 3xI2C, I2S, ADC, DAC, GPIO, JTAG
1.8.2010	Olimex STM32-LCD	STMicroelectronics STM32F103ZE	Cortex M3	72	64KB	512KB					1-otg		LCD, uSD/MMC, 3xUEXT, JTAG
1.8.2010	Olimex STM32-P103	STMicroelectronics STM32F103RBT6	Cortex M3	72	20KB	128KB					1+1-dev		CAN, 2xI2C, 2xADC 12 bit, 3xUART, 2xSPI, GPIO, UEXT, JTAG
19.1.2009	STMicroelectronics STM3210E-EVAL/A	STMicroelectronics STM32F103ZET6	Cortex M3	72	1	16					1		LCD, uSD, 2xRS232, CAN 2.0, I2S, GPIO, JTAG
1.4.2009	Olimex ATMega128 HB	Atmel ATMega128-16AU	AVR-8	16	4KB	128KB							2xUART, SPI, I2C, TWI, ADC, GPIO, JTAG
1.4.2009	Olimex MSP430-P1232M	TI MSP430F1232	MSP430	16	4KB	92KB							2xUART, 2xSPI, 2xI2C, ADC, GPIO, JTAG
1.9.2017	STMicroelectronics STM32F429I-DISCO	STMicroelectronics STM32F429ZIT6	Cortex M4	180	260KB + 8	2	1				1+1-otg		LCD, MEMS, 8xU(S)ART, 6xSPI, 3xI2C, 3xADC, 2xDAC, GPIO, ST-LINK
1.4.2018	STMicroelectronics Nucleo-64 STM32L433	STMicroelectronics STM32L433RCT6PU	ULP Cortex M4	80	64KB	256KB					1		4xUSART, 3xI2C, I2S, 3xSPI, CAN, ADC, 2xDAC, GPIO, ST-Link, Arduino
1.9.2017	STMicroelectronics STMF3-Discovery	STMicroelectronics STM32F303VCT6	Cortex M4	72	48KB	256KB					1		MEMS, 5xU(S)ART, 3xSPI, 2xI2C, 4xADC, 2xDAC, GPIO, ST-LINK
1.1.2010	Olimex Olimexino-STM32	STMicroelectronics STM32F103RBT6	Cortex M3	72	20KB	128KB					1+1-dev		CAN, 2xI2C, 2xADC 12 bit, 3xUART, 2xSPI, GPIO, UEXT, JTAG
1.8.2018	STMicroelectronics Nucleo-64 STM32F072	STMicroelectronics STM32F072RBT6	Cortex M0	48	16KB	128KB					1		4xUSART, 2xI2C, 2xSPI, CAN, ADC, GPIO, On-board ST-Link, Arduino
1.8.2014	STMicroelectronics STM32 Value Discovery	STMicroelectronics STM32F100RBT6B	Cortex M3	24	8KB	128KB							3xUART, 2xI2C, 2xSPI, ADC, GPIO, On-board ST-Link
1.4.2018	Olimex AVR-T32U4 (Leonardo)	Atmel ATMEGA32U4	AVR-8	16	2,56KB	32KB					1-dev		UART, SPI, I2C, TWI, ADC, GPIO
1.4.2018	Olimex Olimexino-32U4	Atmel ATMEGA32U4	AVR-8	16	2,56KB	32KB					1-dev		UART, SPI, I2C, TWI, ADC, GPIO, Arduino layout
1.4.2018	Olimex Olimexino-nano + -bat	Atmel ATMEGA32U4	AVR-8	16	2,56KB	32KB					1-dev		UART, SPI, I2C, TWI, ADC, GPIO, Battery
20.12.2017	Decawave DWM1000 module								802.15.4-2011				UWB 2-way ranging and TDOA, SPI, GPIO, Internal chip antena

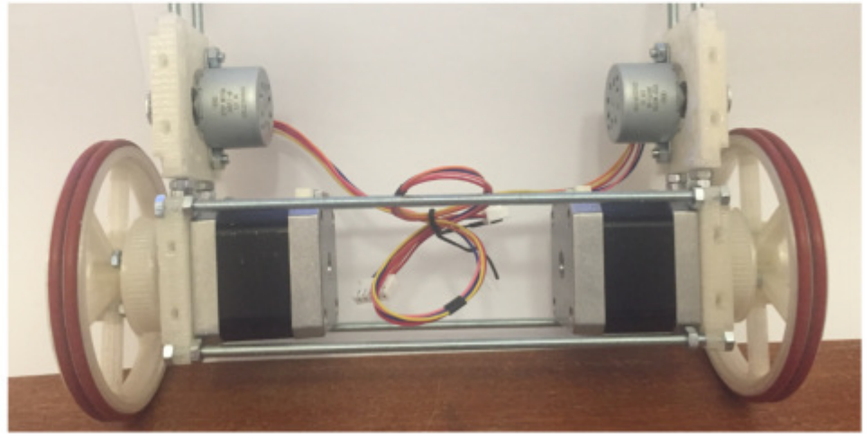
TIOBE index of popularity and programming languages used to gain professional experience

Ranked by TIOBE index of the popularity		TIOBE Index for Dec '20/'19			Long Term History (recorded by TIOBE)								Programming languages used
Programming Language	The Winner of the Year	Dec '20	Monthly Change	Dec '19	2020	2015	2010	2005	2000	1995	1990	1985	Used in various projects (Year, CPU, OS, SW, Usage)
C	'19, '17, '08	1	^	2	1	2	2	1	1	1	1	1	'83-'20, MC6800/MOS 6502/MC680x0/x86-64/x51/AVR-8/SAM9260/EP9302/AVR-32/E1-32/ARM Cortex M3/M4/A7/A8, FW/MDOS/MS-DOS/Windows/OS-9/Unix/QNX/Linux/eCOS/uSmartOS/FreeRTOS/ChibiOS
Java	'15, '05	2	v	1	2	1	1	2	3	29	-	-	'03-'20, x86-64/MIPS32/Xscale/TC65/E1-32/ARM Cortex A7/A8, FW/Windows/Mac-OS/Linux, OSGI/Eclipse/Web staff
Python	'20, '18, '10, '07	3	-	3	3	5	6	7	22	13	-	-	'03-'20, x86-64/MIPS32/Xscale/ARM Cortex A7/A8, Windows/Linux, Web staff
C++	'03	4	-	4	4	3	3	3	2	2	2	8	'00-'01, R4000/ERS-210, Aperios, FW for AIBO robots/RoboCup 2001; '00-'20, x86-64, Windows/Linux, C++ Builder/Eclipse; '10-'20, Atmel AVR 8bit/AtMega32U4/ARM Cortex M3/M4, FW, Arduino
C#	-	5	-	5	5	4	5	6	10	-	-	-	'03-'08, x86, Windows, Web staff; '10-'11, Teklogix 7530, Windows CE
Visual Basic	-	6	-	6	-	-	-	-	-	-	-	-	'00-'20, x86-64, Windows, MS Office, Scientific Calculations
JavaScript	'14	7	-	7	6	8	10	10	7	-	-	-	'15-'20, x86-64/ARM Cortex A7/A8, Windows/Mac-OS/Linux, NodeJS/Web staff
PHP	'04	8	-	8	7	6	4	4	19	-	-	-	'07-'20, x86/MIPS32/Xscale/ARM Cortex A7/A8, Windows/Linux, Web staff
R	-	9	^^	16	9	14	46	-	-	-	-	-	-
SQL	-	10	-	9	8	-	-	-	-	-	-	-	'95-'03, x86, Windows/NT, Sybase SQL/Delphi/C++ Builder; '07-'20, x86/MIPS32/Xscale/ARM Cortex A7/A8, MySQL/Windows/Linux, .NET/PHP/Web staff
Groovy	-	11	^^	22	-	-	-	-	-	-	-	-	-
Assembly language	-	12	^	14	-	-	-	-	-	-	-	-	'79-'89, MC6800/MOS 6502, FW/MDOS; '84-'89, LSI 1123, RT-11; '88-'89, MC680x0, OS-9/Unix; '95-'96, x86, MS-DOS; '03-'08, E1-32, HyNetOS/Linux
Swift	-	13	v	10	10	15	-	-	-	-	-	-	-
Perl	-	14	^^	20	-	-	-	-	-	-	-	-	-
Ruby	'06	15	vv	11	-	-	-	-	-	-	-	-	-
Go	'16, '09	16	v	15	-	-	-	-	-	-	-	-	-
MATLAB	-	17	-	17	-	-	-	-	-	-	-	-	-
Delphi/Object Pascal	-	18	vv	12	-	-	-	-	-	-	-	-	'94-'03, x86, Windows, Delphi, Desktop Applications
Objective-C	'12, '11	19	vv	13	-	-	-	-	-	-	-	-	-
PL/SQL	-	20	^^	24	-	-	-	-	-	-	-	-	-
Transact-SQL	'13	25	-	-	-	-	-	-	-	-	-	-	-
Lisp	-	27	-	-	29	26	14	13	9	6	4	2	'86-'89, MC680x0, OS-9/Unix; '87-'93, x86, MS-DOS/Windows, Acad
Fortran	-	32	-	-	31	21	24	14	13	14	3	5	78-'79, IBM System 360, Punched card OS, Scientific Calculations
Ada	-	34	-	-	34	23	21	16	17	3	9	3	-
Lua	-	43	-	-	-	-	-	-	-	-	-	-	'14-'18, x86-64, Windows, FEMM, Scientific Calculations
Basic	-	-	-	-	-	-	-	-	-	-	-	-	'83-'89, Apple II, ROM/FDD, Scientific Calculations
Occam	-	-	-	-	-	-	-	-	-	-	-	-	'89-'93, T414/T800, -/Helios, Own HW/SW Development System
Parallel C	-	-	-	-	-	-	-	-	-	-	-	-	'89-'93, T414/T800, -/Helios, Own HW/SW Development System
Parallel Fortran	-	-	-	-	-	-	-	-	-	-	-	-	'89-'93, T414/T800, -, Scientific Calculations
Focal	-	-	-	-	-	-	-	-	-	-	-	-	'78-'79, CM-4, Punched tape OS, Scientific Calculations

Because of the long professional life and great number of use cases used programming languages are compared to the index of popularity calculated by TIOBE. As can be seen from the table, nine of the ten most popular programming languages were used back in time. Another ten used programming languages fall among the top fifty of TIOBE's lists for the year 2020. On the other hand almost all 20 programming languages used in time fall among the top fifty of TIOBE's lists for the last 35 years.

Own project: Adroid the Open S.T.E.A.M. robot platform (2024)





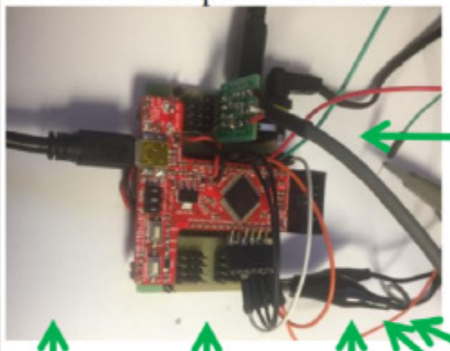
Own project: Multichannel Energy Metering (MEM) system and its test bench (2023)

Option 2

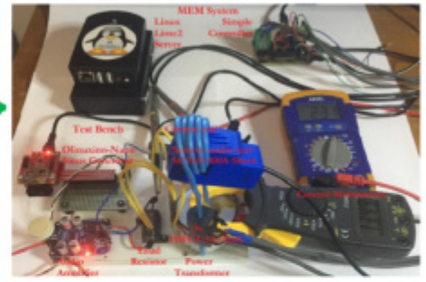
Olimexino-Nano-BAT

12V/4A AC-DC adapter
or 4S LiPo Battery & Charger
and 5V DC-DC converter

MEM simple controller



Lime2-Server-128GB-SDD



Option 1

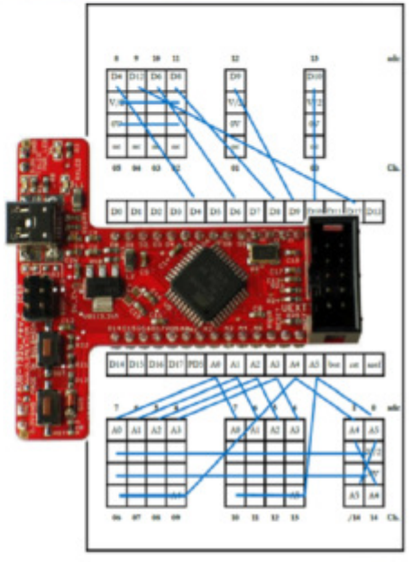
Olimexino-Nano-BB

ZK-4KX DC-DC or other
22V/4A step up convertor

Up to 12 HWCT-5A/5mA

SCT013 100A 50mA

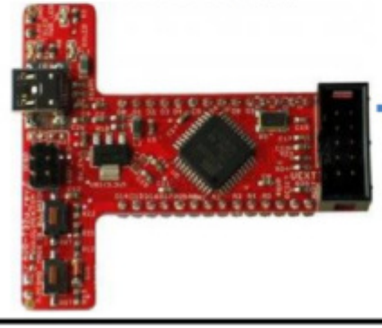
ANG 6V/1A AC-AC adapter



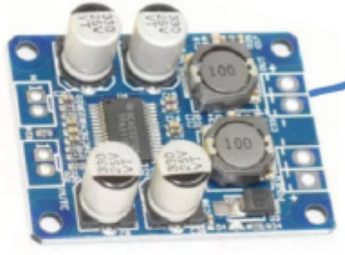
Base test bench staff

Olimexino-Nano

or AVR-T32U4

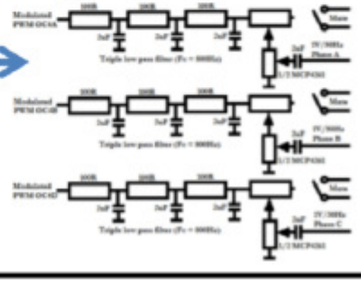


TPA3118 based class D
audio amplifier (mono)



Up to 3x boards

Triple low pass filters,
voltage and current control



Up to 3x INDEL TST 10/006
10VA/230V/8V/1.25A or

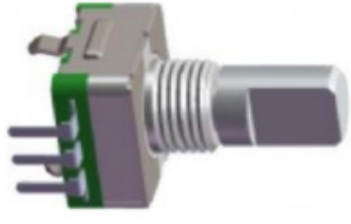
ANG special 6VA transformer

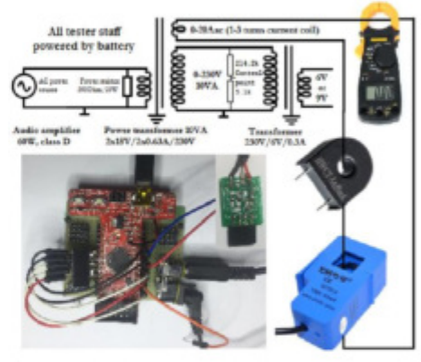
3.2 inch color 240x320 pixels
TFT display module

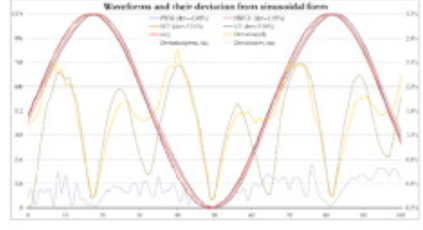
PZEM-022 AC multi-meter

or ATM90E32AS metering

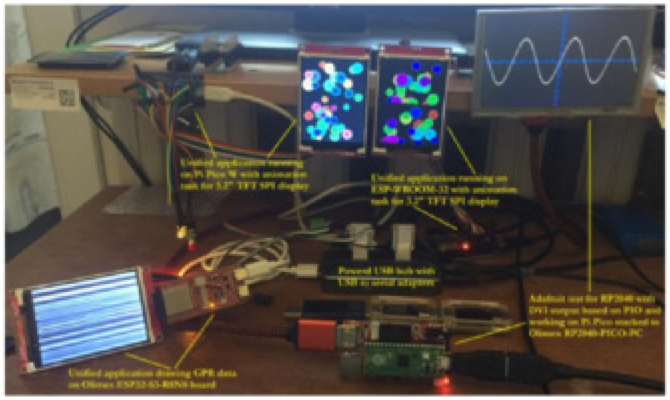
Rotary encoder with push-on
switch ED1115S-20-20F

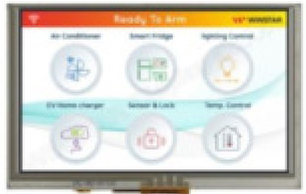


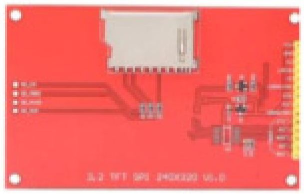


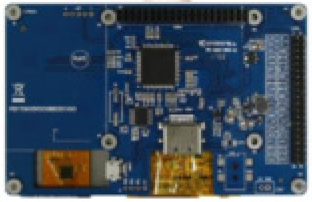


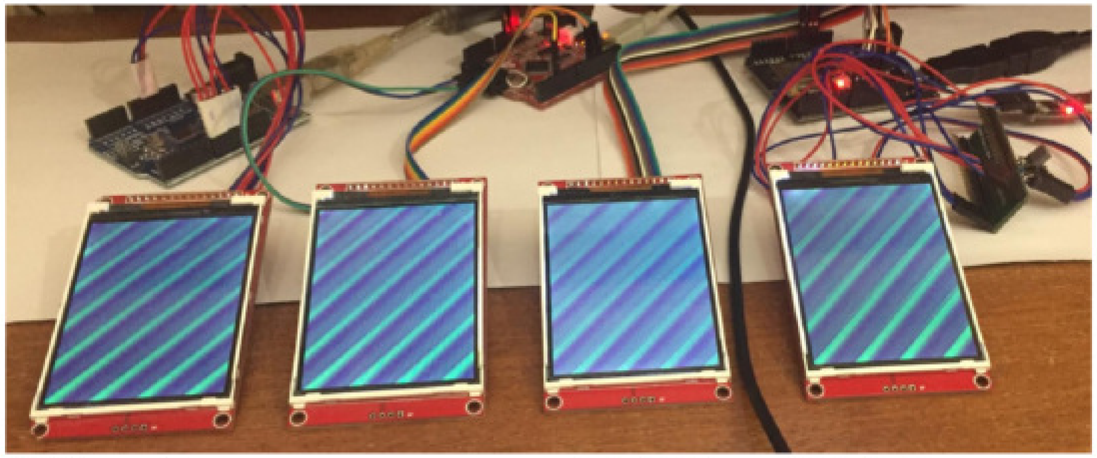
Own project: Unified Multicore Low Power IoT Platform (2023)

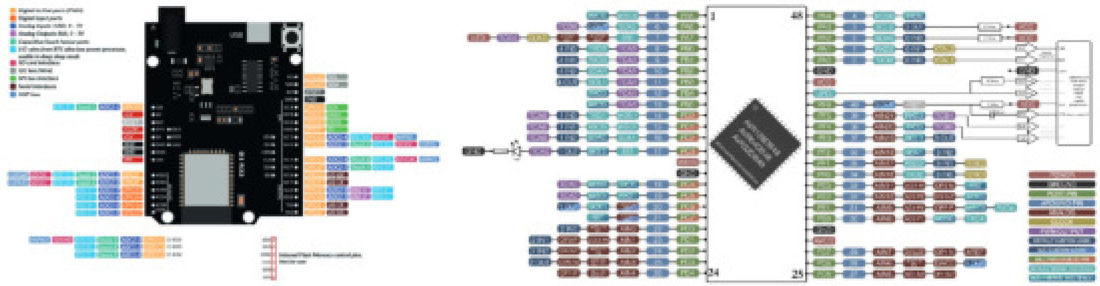


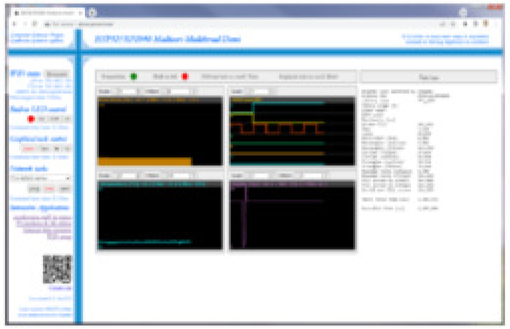


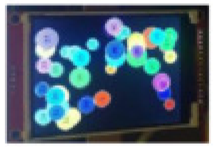


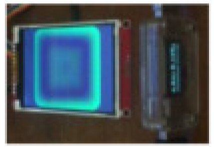


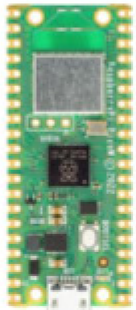


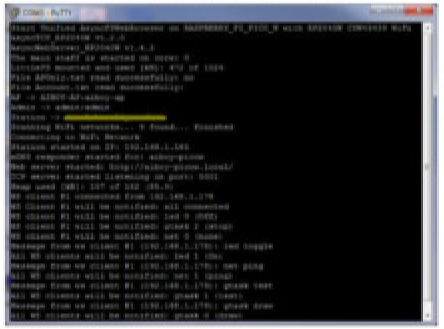


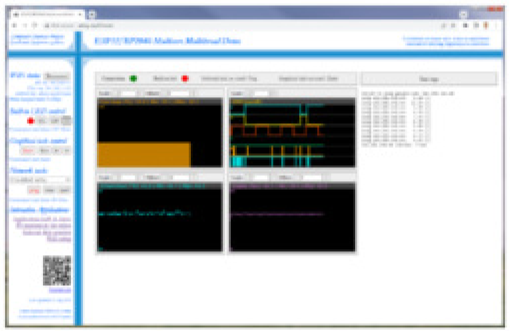


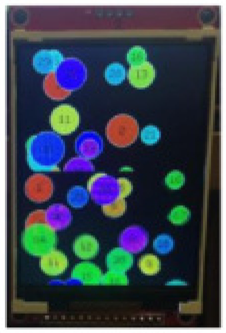


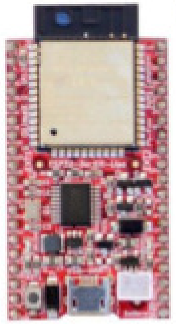


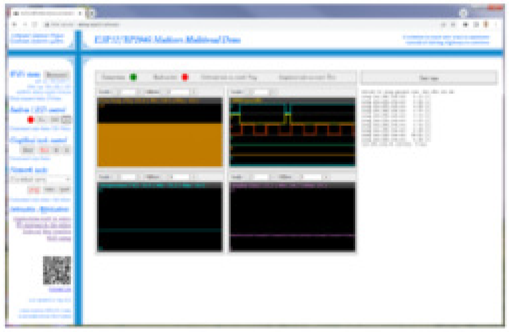


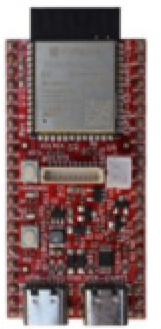


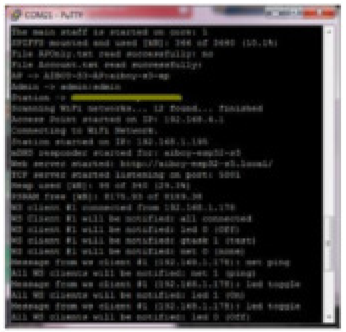






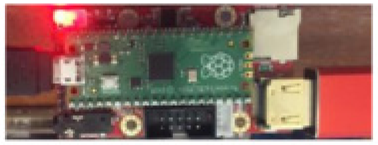


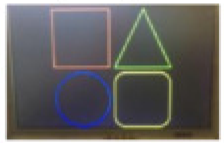


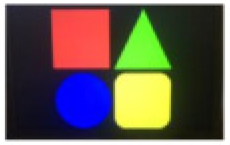


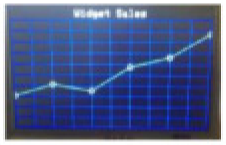
Arduino board / MCU	UNO / ATMega328			Leonardo / ATMega32u4			D1 R32 / ESP32			Pi Pico / RP2040			Pi Pico / RP2040 (Overclocked)			Unified App	AVR128db48	ESP32-S3
ILI9341 Library used (SPI clock)	Adafruit	TFT	Speed up	Adafruit	TFT	Speed up	Adafruit (3MHz)	TFT_eSPI	Speed up	Adafruit	TFT_eSPI (27MHz)	Speed up	Adafruit	TFT_eSPI (62.5MHz)	Speed up	TFT_eSPI with DMA	Adafruit	Adafruit (27MHz)
Memory usage [B]																		
Flash used	23,736 of 32,256 (73.59%)	21,870 of 32,256 (67.80%)		25,874 of 28,672 (90.24%)	23,992 of 28,672 (83.68%)		237,600 of 1,310,720 (18.13%)	295,261 of 1,310,720 (22.53%)		327,772 of 2,093,056 (15.66%)	372,092 of 2,093,056 (17.78%)		327,868 of 1,568,768 (20%)	372,180 of 1,568,768 (23%)		505,232 of 1,044,480 (48%)	24,154 of 130,560 (18.65%)	295,261 of 1,310,720 (22.53%)
SRAM used	950 of 2,048 (46.39%)	746 of 2,048 (36.43%)		915 of 2,560 (35.74%)	711 of 2,560 (27.77%)		37,264 of 327,680 (11.37%)	19,480 of 327,680 (5.94%)		71,324 of 262,144 (27.21%)	71,768 of 262,144 (27.38%)		71,328 of 262,144 (27%)	71,768 of 262,144 (27%)		74,912 of 262,144 (28%)	1,087 of 16,384 (6.63%)	19,480 of 327,680 (5.94%)
Benchmarks [us]								~42°C			~32°C			~34°C		~36°C		~50°C
Screen fill	1,496,456	870,220	1.720	1,503,590	874,600	1.720	2,120,993	274,575	9.097	604,056	281,577	2.145	497,451	107,972	4.607	107,567	1,603,604	274,575
Text	147,088	60,416	2.435	147,820	60,724	2.434	99,630	32,599	6.498	48,452	18,831	2.414	30,599	8,085	3.785	8,070	114,885	32,599
Lines	1,172,116	242,732	4.829	1,178,004	243,588	4.826	986,746	339,491	10.973	454,856	101,897	4.464	304,234	42,740	7.118	43,648	946,199	339,491
Horiz/Vert Lines	125,064	71,336	1.753	125,656	71,696	1.753	173,171	24,171	8.603	50,042	23,541	2.126	40,853	9,078	4.500	8,880	132,637	24,171
Rectangles (outline)	82,228	45,844	1.794	82,632	46,076	1.793	110,682	15,996	8.697	32,657	14,932	2.187	26,417	5,773	4.576	5,686	85,703	15,996
Rectangles (filled)	3,107,069	1,807,436	1.719	3,122,844	1,816,740	1.719	4,402,687	570,510	9.096	1,253,856	584,372	2.146	1,032,576	228,086	4.608	223,506	3,329,307	570,510
Circles (filled)	452,728	284,064	1.594	454,956	285,536	1.593	492,735	95,609	7.704	167,554	71,149	2.366	126,969	28,025	4.531	27,896	423,221	95,609
Circles (outline)	497,252	135,580	3.668	499,604	136,148	3.670	432,728	150,143	12.978	199,626	37,258	5.358	133,263	15,560	8.566	15,743	404,412	150,143
Triangles (outline)	261,036	59,496	4.388	262,392	59,808	4.387	225,099	74,839	10.263	101,400	23,636	4.290	68,473	10,319	6.636	10,463	213,681	74,839
Triangles (filled)	1,330,720	694,456	1.916	1,337,200	698,032	1.916	1,432,757	209,558	8.698	429,598	195,995	2.194	345,244	75,450	4.576	75,102	1,279,412	209,558
Rounded rects (outline)	228,892	100,004	2.289	230,024	100,532	2.288	230,767	62,675	11.013	92,290	23,635	3.904	65,233	9,576	6.812	9,602	209,582	62,675
Rounded rects (filled)	3,127,968	1,976,936	1.582	3,143,588	1,987,180	1.582	4,394,111	378,680	8.995	1,257,871	586,292	2.145	1,032,024	225,027	4.586	224,252	3,330,731	378,680
Fill screen by pixels	3,360,992	918,732	3.658	3,387,508	923,492	3.668	2,763,609	1,591,181	3.331	1,255,234	504,753	2.487	805,373	229,256	3.513	159,327	2,964,699	1,591,181
Fill screen by bitmaps	528,576	855,088	0.618	531,112	859,520	0.618	435,205	62,752	0.518	66,436	520,180	0.128	70,363	234,904	0.300	166,052	453,099	62,752
Scroll and fill screen	**532,988**	**855,696**	**0.623**	**535,808**	**860,152**	**0.623**	**439,860**	**67,668**	**0.520**	**69,357**	**521,011**	**0.133**	**71,933**	**235,385**	**0.306**	**166,606**	**457,946**	**67,668**
Min	82,228	45,844		82,632	46,076		99,630	15,996		32,657	14,932		26,417	5,773		5,686	85,703	15,996
Avg	**1,097,346**	**598,536**	1.833	**1,102,854**	**601,634**	1.833	**1,250,108**	**276,722**	4.497	**405,402**	**233,937**	1.733	**310,067**	**97,416**	3.185	**83,496**	**1,062,687**	**276,722**
Max	3,360,992	1,976,936		3,387,508	1,987,180		4,402,687	1,591,181		1,257,871	586,292		1,032,576	235,385		224,252	3,330,731	1,591,181
Sum	16,460,184	8,978,036		16,542,808	9,024,204		18,751,620	4,150,627		6,081,037	3,509,059		4,651,005	1,461,240		1,252,440	15,940,298	4,150,627
DrawWithDMA test (bouncing of 42 colored and numbered circles)								30fps (Unified App)			17.8 fps at CPU 133MHz SPI 27MHz			46.5 fps at CPU 250MHz SPI 62.5MHz	2.6	46.5 fps at CPU 250MHz SPI 62.5MHz		60+ fps at GPR data visualisation

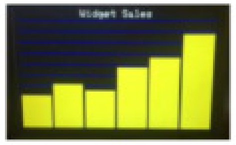
Full color PicoDVI test on Raspberry Pi Pico, RP2040-PICO-PC and 5” TFT HDMI 800x480 display

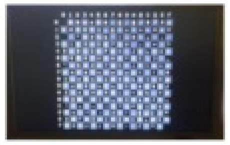


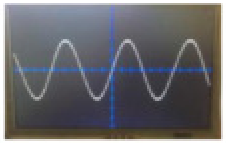


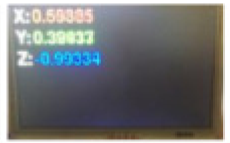




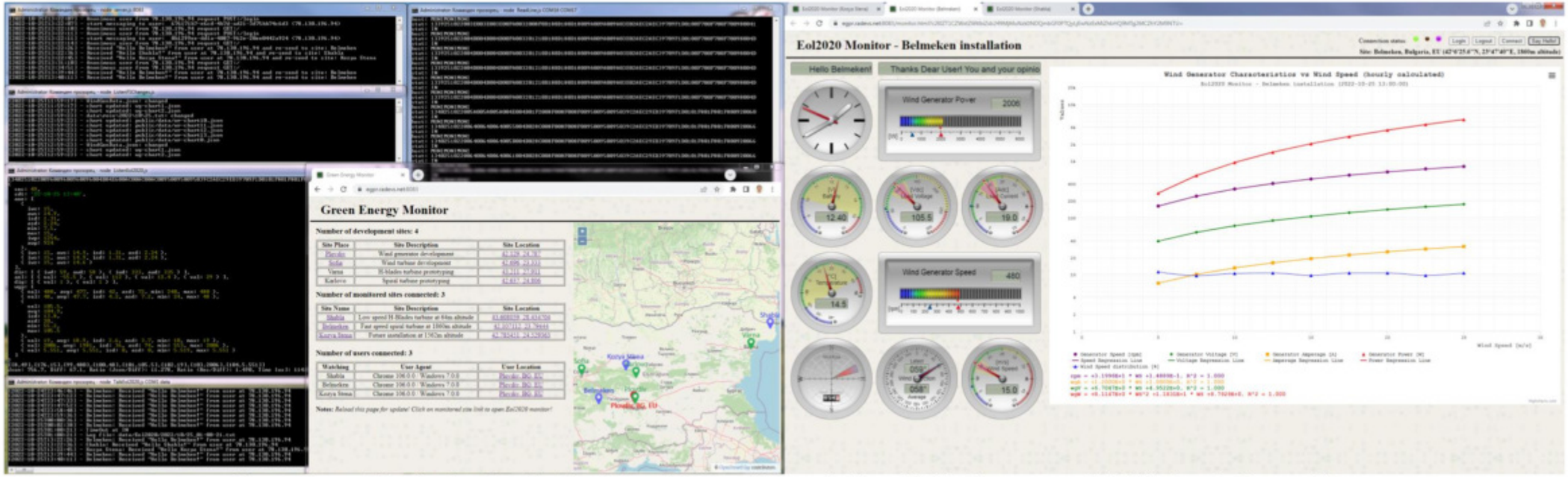




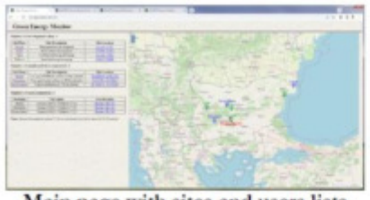




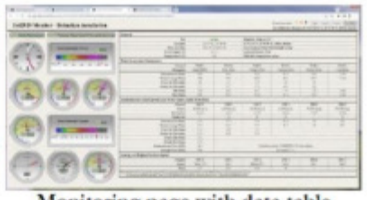
Own project: Green Energy Monitoring Project (2020 – 2022)



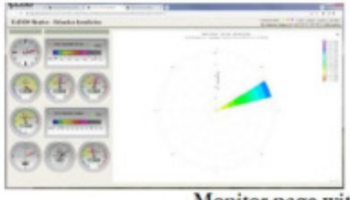
Green Energy Monitoring Application (Https/WSS multithreaded local/global servers and HTML 5 web client staff)



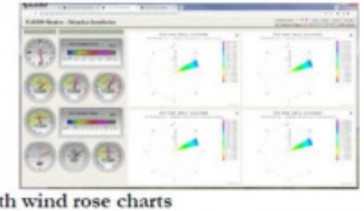
Main page with sites and users lists

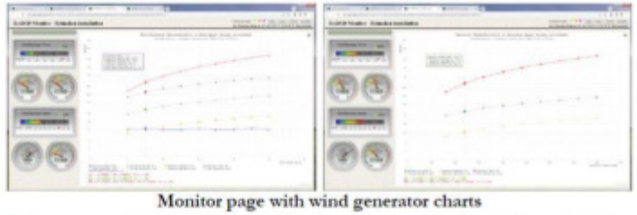


Monitoring page with data table

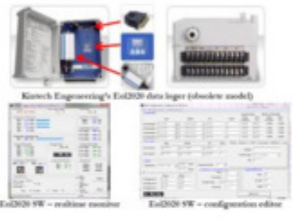


Monitor page with wind rose charts



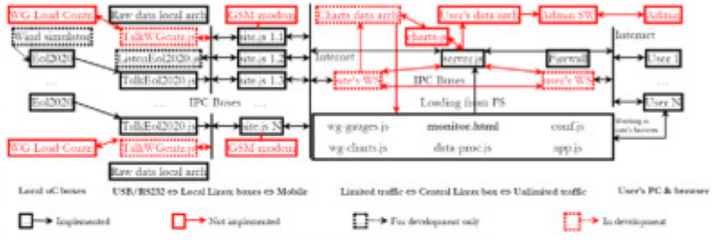


Monitor page with wind generator charts



Eol2020 data logger and weather sensors

Eol2020 serial protocol hacking

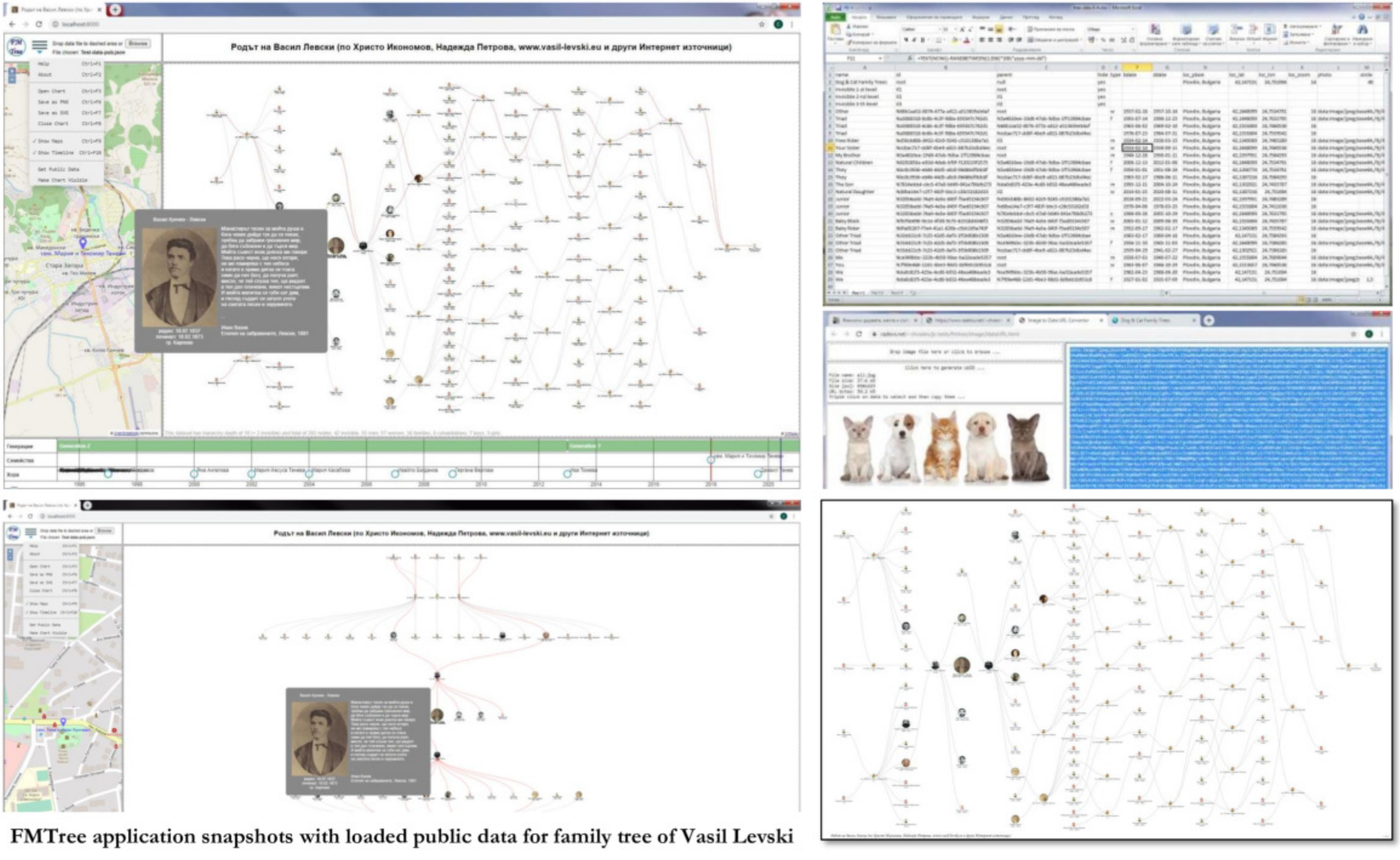


System staff schematics

The project is started as supporting staff for the field installations of the wind turbines. The main idea was to capture weather information synchronously with the data from the wind turbines working at different places on the field where Kintech Engineering's Eol2020 data loggers were installed. In addition a special MPPT controller was planned to operate the wind turbine and/or photovoltaic load. It was developed appropriate instrumentation and methodology for hacking Eol2020 serial protocol. As a result from successful work a new monitoring system was designed. It is based on Secure Web Sockets, Https, Express middleware, Node IPC and consists of Node JS multithreaded local and global servers and HTML 5 client web application. There are plenty of JS libraries responsible for system functioning but some of the most important once used for data processing and visualization are OpenLayers framework & OpenStreetMap interactive maps, Highcharts JS charting framework, Regression-JS, SteelSeries Gauges and Rose Gauges. Captured information from multiply installation sites is stored, processed and visualized in table and chart views. It can be monitored by many users simultaneously and recast later on.

Own project: Listen and play piano - web application for children (2021)

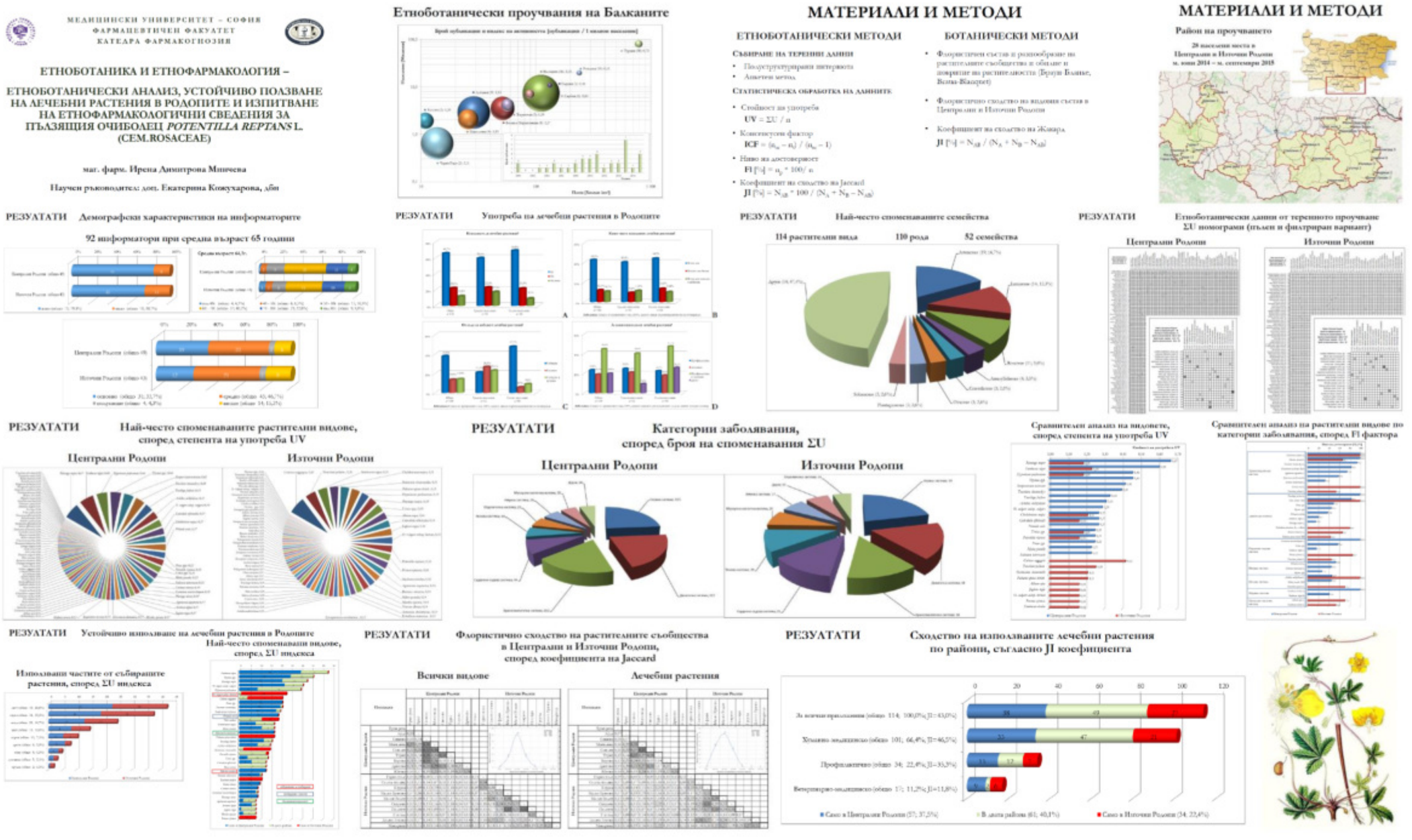
Own project: Family Multi-Tree web application (2019 – 2020)



FMTree application snapshots with loaded public data for family tree of Vasil Levski

This software is implemented thanks to Data-Driven Documents (d3.js) library, OpenLayers dynamic maps framework, OpenStreetMap foundation & contributors, d3.js timeline chart by Genscape Inc. and many other libraries and demos. It is written with HTML 5, JavaScript, CSS and works well on almost all modern browsers, locally with data file loaded directly and globally via web server over Internet. Private data can be secured by using local files only or protected Internet place. The application uses CSV or JSON formatted data files to present advanced multitree charts. In addition to the chart, interactive world map and timeline are presented as well. An application for converting images to Data URL is also developed to help the users to deal with text only data. Final multitrees can be exported as high resolution images for publishing and/or printing.

Doctoral dissertation: mathematical support, paper and presentation design and work out (2018 – 2019)
Subject: Ethnobotany and ethnopharmacology – ethnobotanical analysis, sustainable use of medicinal plants in the Rhodopes and testing ethnopharmacological data on the reptile Potentilla reptans L. (family Rosaceae)



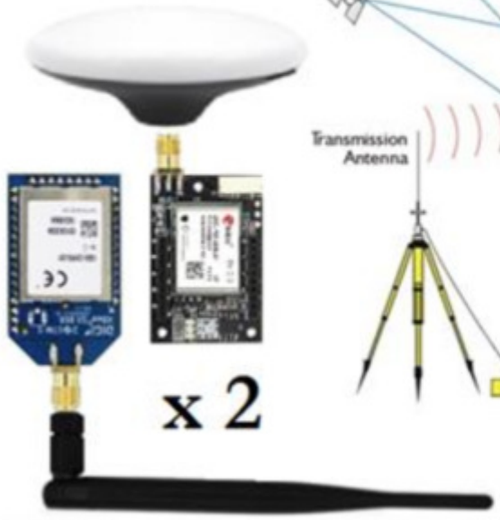
The assistance provided is in the final phase of the preparation of the doctoral dissertation in the field of ethnobotany and ethnopharmacology. It was expressed in the mathematical processing of the data, their presentation as tables and charts, the preparation of the final print out, the design and work out of the printed abstract and the presentation. In addition, a new method for presenting the data in the form of nomograms was proposed and implemented. Assistance in the preparation of several articles was also provided later on.

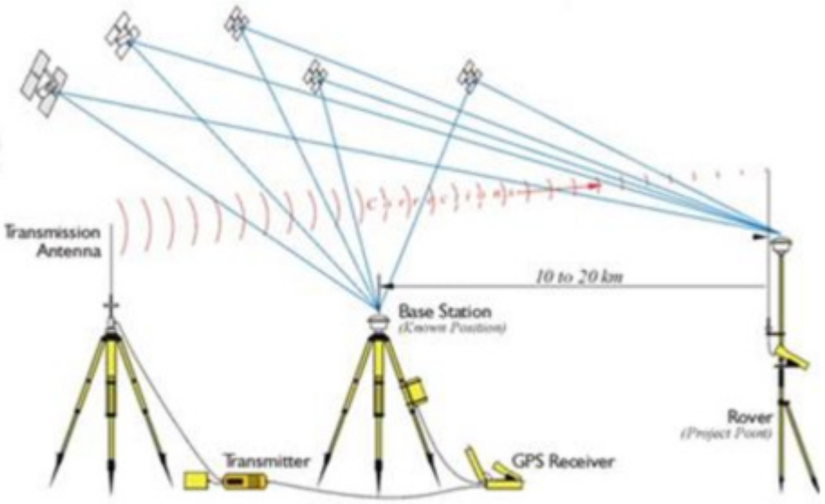
Work in progress: GPS/LPS solution for EGPR (2015 – 2020)

Rover staff

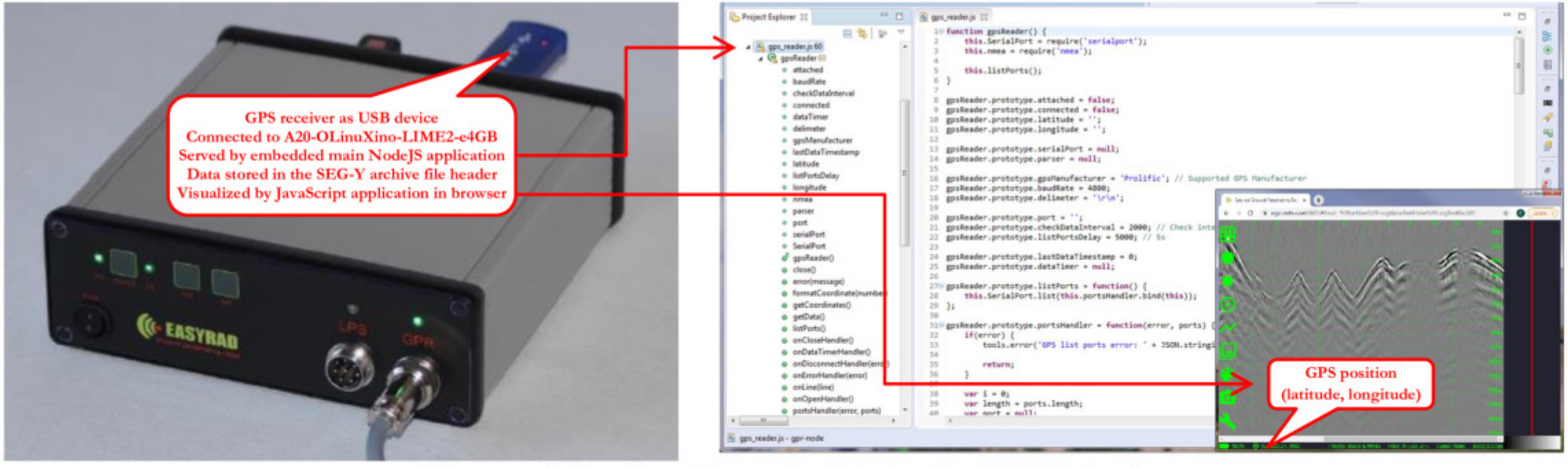
Base control staff

Mobile EGPR platform with precise positioning and control



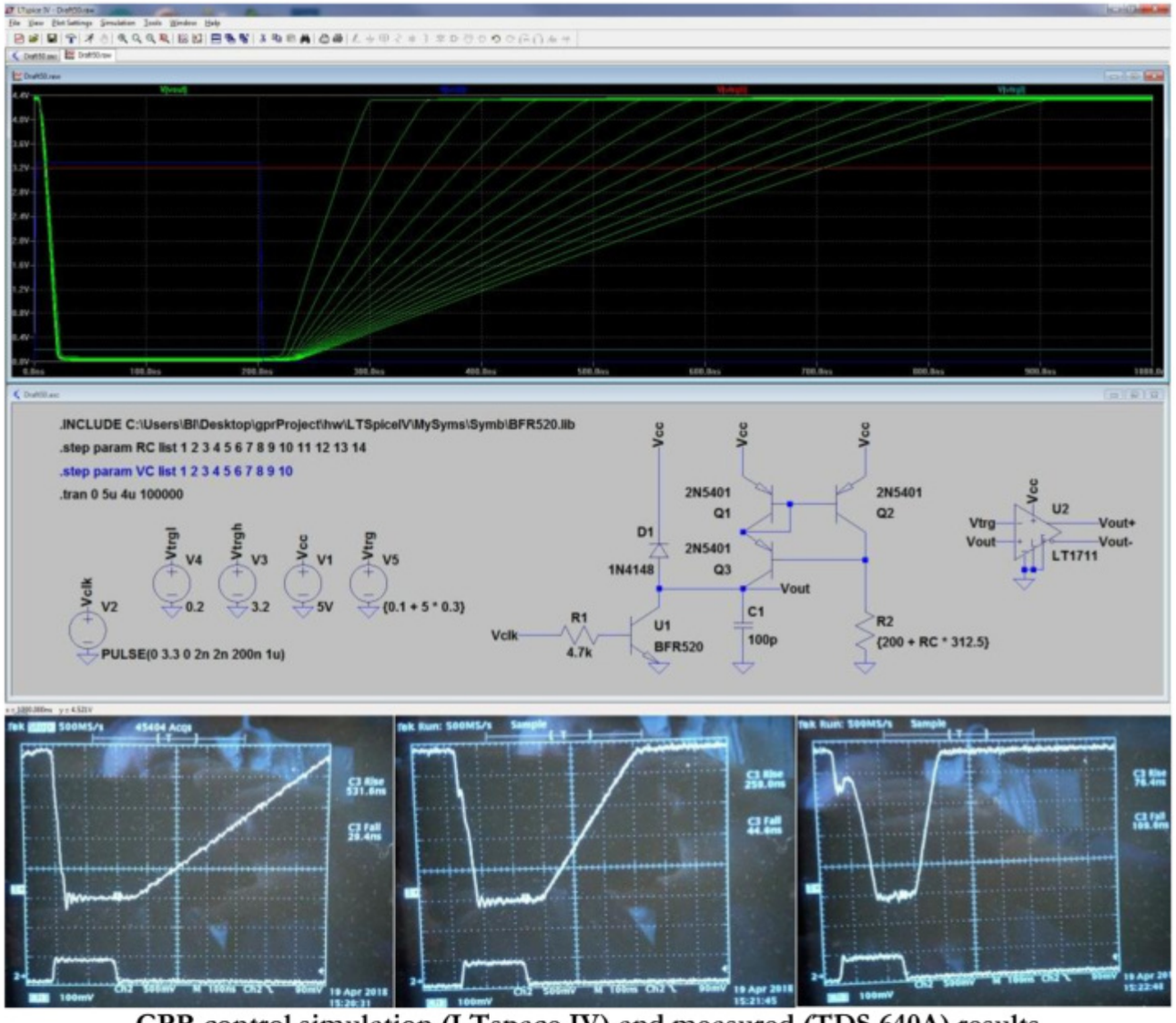


High precision positioning based on RTK GPS modules like ZED-F9P of U-blox

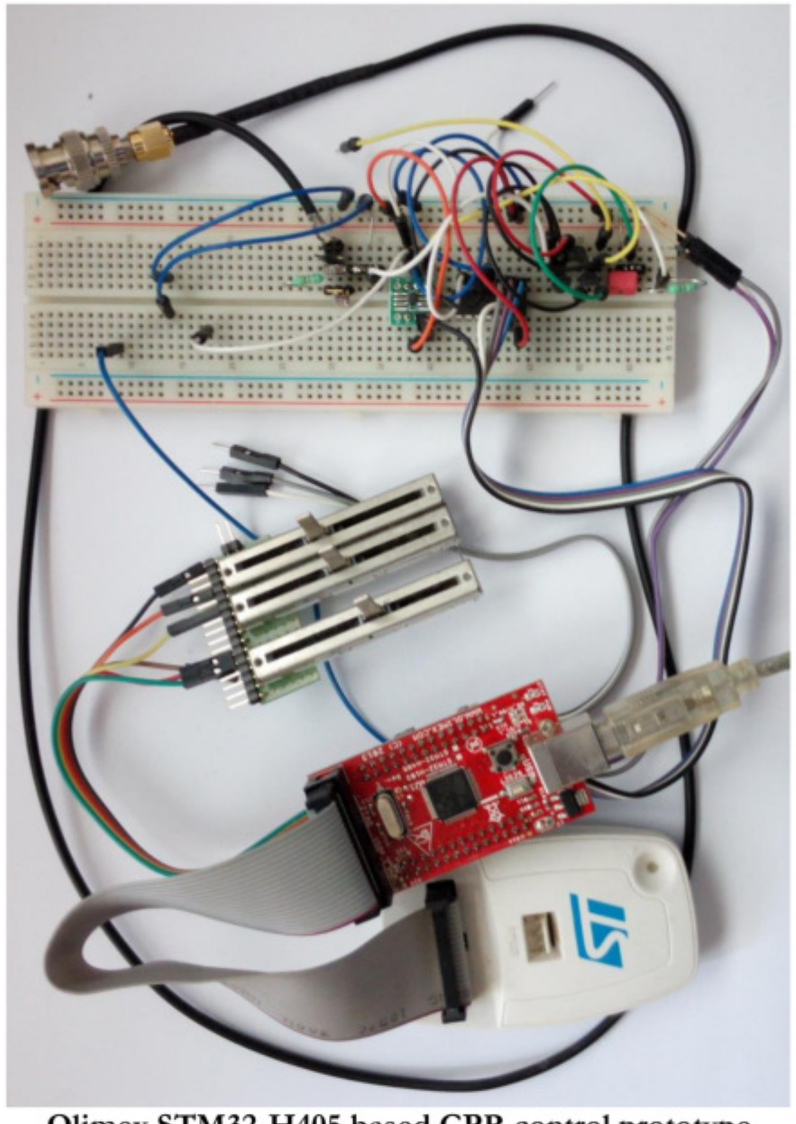


GPS support for Easy Ground Penetrating Radar based on USB GPS dongle

One of the latest implementations is related to the GPS localization based on a USB GPS dongle and data written in the SEG-Y files. As a consequence of this, the idea of precise positioning was further developed towards the use of RTK GPS based on modules like ZED-F9P of U-blox. The idea of mounting of the GPR on a remote-controlled mobile device was also floated. For better vehicle control stereo camera and FPV Goggles are proposed. The camera vision can be combined with EGPR application view in the Linux box and broadcasted for observation by many people. Optionally the video can be saved for later review.



GPR control simulation (LTspace IV) and measured (TDS 640A) results

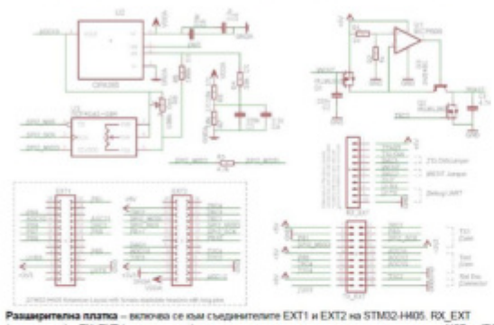


Olimex STM32-H405 based GPR control prototype

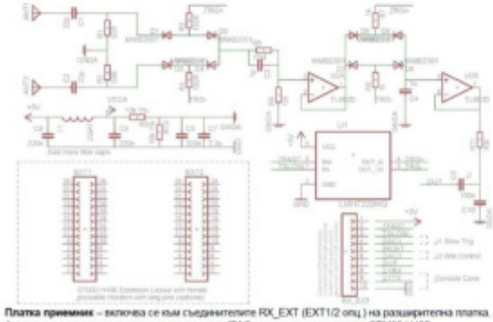
The GPR control solution was investigated and couple of implementations were tested. The latest and the best one was simulated in LTspace IV, prototyped and tested with 500MHz TDS 640A oscilloscope. The simple hardware and the good software control implemented on STM32F405 MCU makes it possible to create extremely flexible and really easy controlled GPR even multi frequency one.

Work in progress: Multi-frequency solution for EGPR (2015 – 2020)

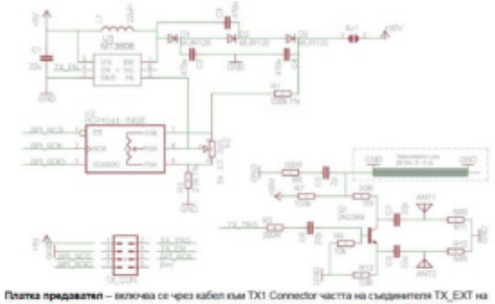
Схеми на вариант 4 – сандвич от STM32-H405 и 2(3) платки 61x34 mm



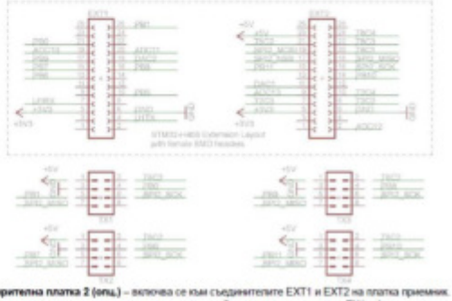
Разширителна платка – включва се към съединителите EXT1 и EXT2 на STM32-H405. RX_EXT (вертикален) и TX_EXT (хоризонтален) съединителите се монтират по тясната страна под USB и JTAG съединителите на STM32-H405 съответно.



Платка приемник – включва се към съединителите RX_EXT (EXT1/2 опц.) на разширителна платка. Антените се монтират по тясната страна под JTAG съединителя на платка STM32-H405.

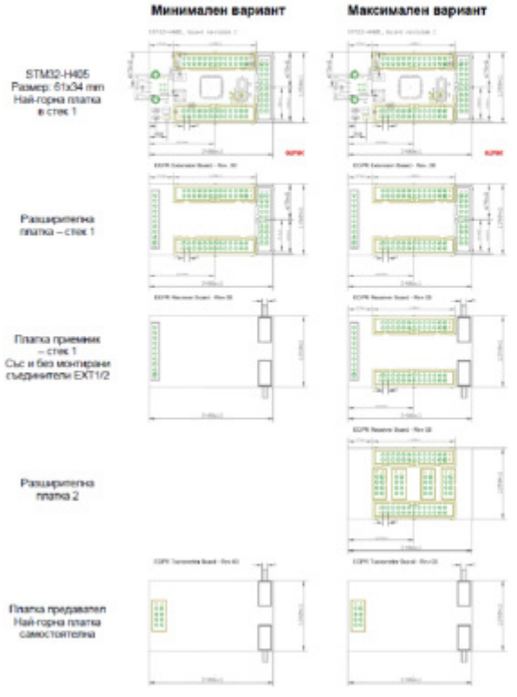


Платка предавател – включва се чрез кабел към TX1 Connector частта на съединителя TX_EXT на разширителна платка. Антените се монтират по тясната страна срещуположна на TX_CON. Закъснителната линия към платка предавател може да се използни или външу платка или чрез коаксиален или UTP кабел с подходящи характеристики. При нужда от използване на закъснителна линия и когато е реализирана на платка тя може да е със същите размери и да се монтира във вид на сандвич с платката предавател. При нужда от по голяма дължина на закъснителната линия може да се ползва или кабел или повече от 1 платка с линии свързани последователно.

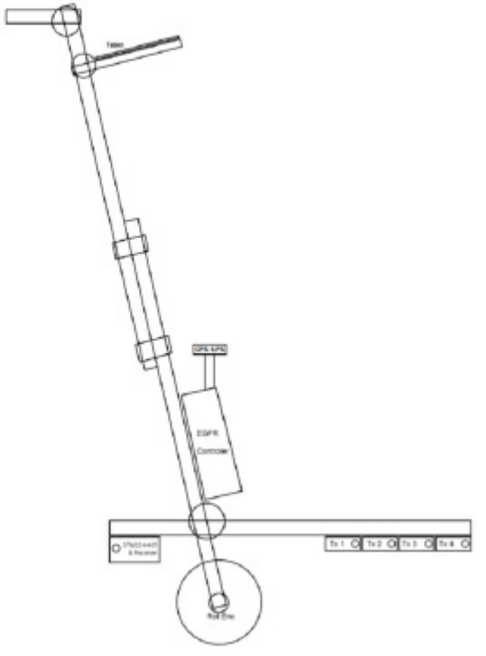


Разширителна платка 2 (опц.) – включва се към съединителите EXT1 и EXT2 на платка приемник. До 4 платки предавател могат да се включат чрез кабели към съединителите TX1...4. За свързване може да се използва Ethernet кабел с усукани двойки и съединители CAT3.

Платки на вариант 4 – размери и оформление

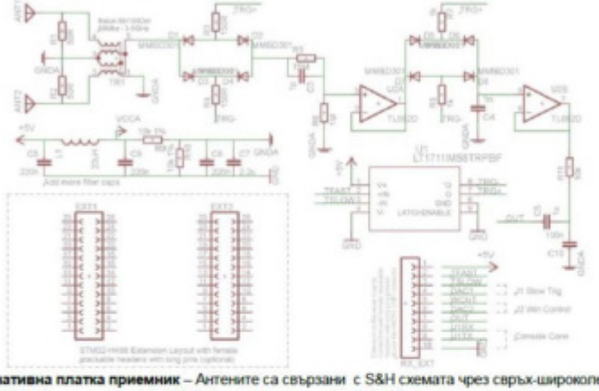


EGPR – изглед на стойката и разположение на елементите

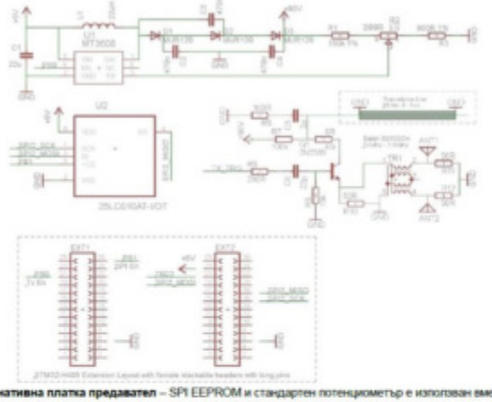


Варианти на предавателя: 50, 200 и 350 MHz за подземни обекти с диполна антена;
500, 750 и 1000 MHz за подземни обекти с екранирана антена;
1500 MHz за бетон с екранирана антена

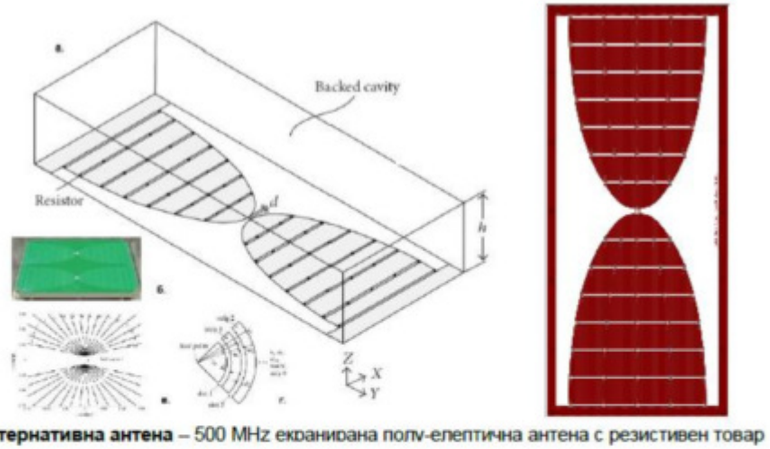
Алтернативни схеми на вариант 4



Алтернативна платка приемник – Антените са свързани с S&H схемата чрез свръх-широколентов (20-5500 MHz) съгласуващ (50-100 Om) трансформатор (балун). За бърз компаратор се ползва LT1711 (Rail-to-Rail Complementary output, 2 ns Output Rise / Fall time and 4.5 ns Propagation Delay).



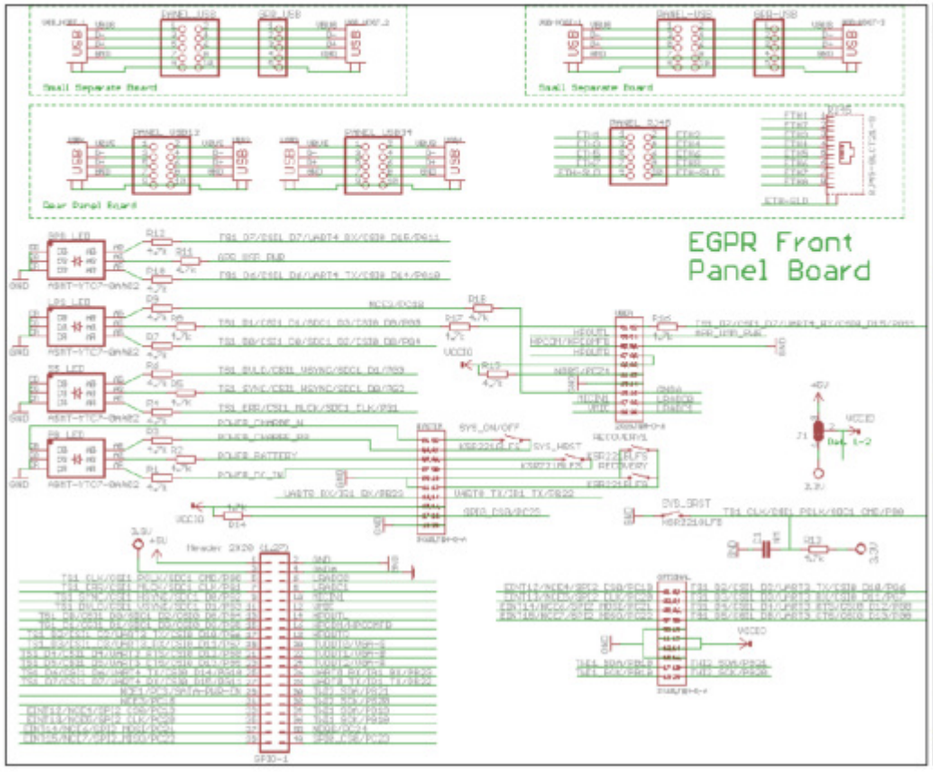
Алтернативна платка предавател – SPI EEPROM и стандартен потенциометър е използван вместо дигитален такъв. Антените са свързани с изхода на генератора чрез свръх-широколентов (20-5500 MHz) съгласуващ (50-100 Om) трансформатор (балун). Платката има STM32-H405 EXT конектори за монтаж в общ стек с другите платки.

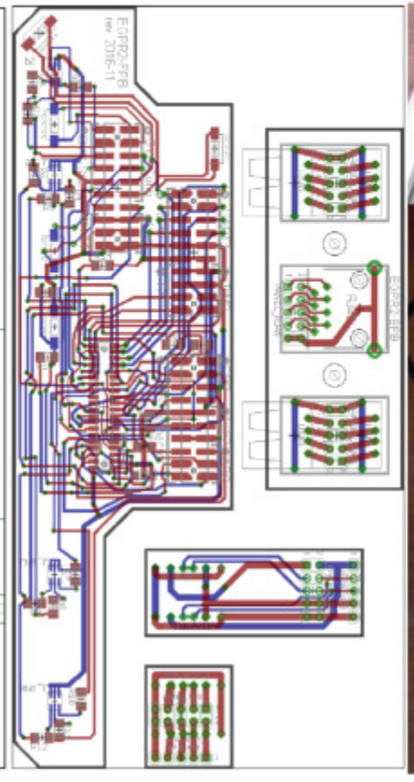


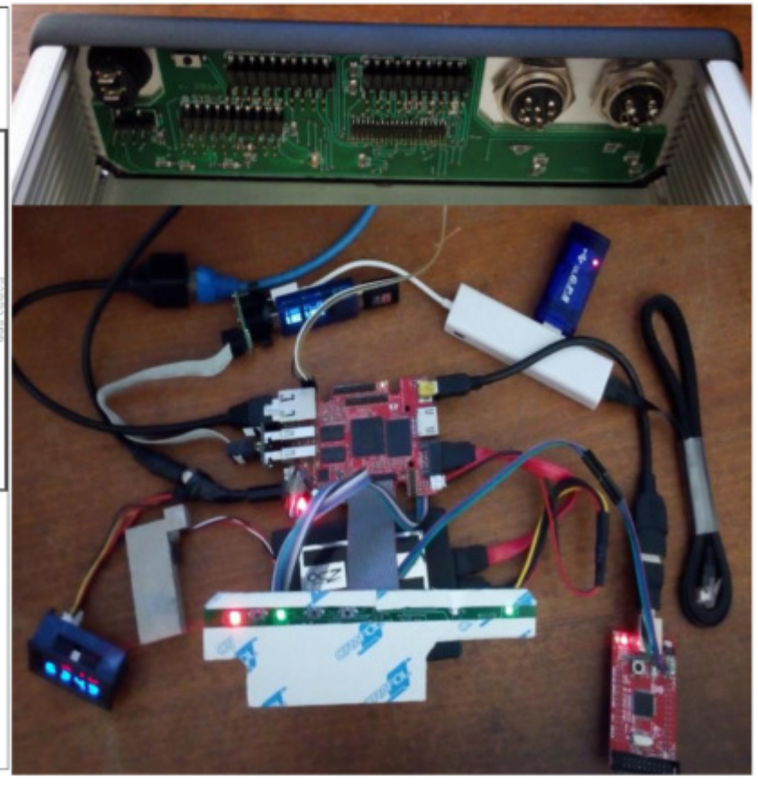
Алтернативна антена – 500 MHz екранирана полу-елептична антена с резистивен товар

Multi frequency GPR solution with dipole antennas (up to 300 MHz) was investigated, implemented and tested. It was found a couple of problems like the quality of the saw tooth signal and big noise in the receiver stage. Nevertheless, mechanical solution was assessed as good enough for the later usage. In addition, a 500MHz semi-elliptical antenna with a resistive load was investigated and evaluated as a very good solution for a 500 MHz GPR. Meanwhile, the possibility of short pulse generation directly from the MCU was explored. Several candidates manufactured by ST Microelectronics were identified like STM32F334 and STM32G474 having timers with picosecond resolution (217ps and 184ps respectively). Both they are Cortex-M4 32-bit MCU+FPU working up to 72 MHz and 170 MHz respectively. The disadvantages are the low performance of STM32F334 and the difficult delivery of STM32G474 chips in small quantities. Other problem that should be solved is the signal amplification to reach required power.

Own project: Easy Ground Penetrating Radar (2015 – 2018)
Complex co-design: Components, interconnections and casing

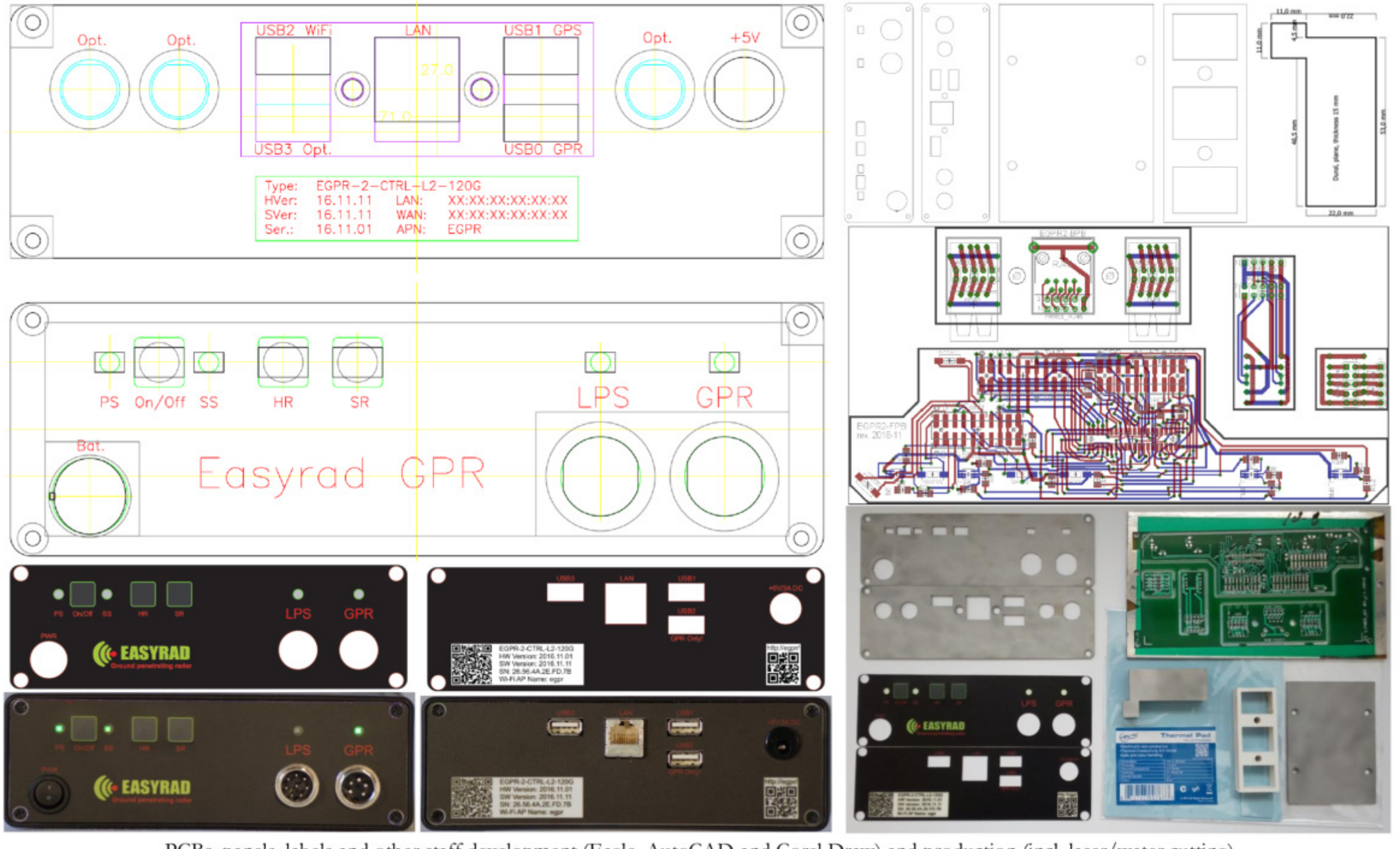






Two Linux boxes were finally assembled. Serious attention was also paid to the heat transfer in order to prevent overheating of the interior. The software was built and installed. Final long term high load functional tests was executed and finished successfully. The application controller was connected to EASYRAD's GPR and all the system was tested and found to be working as expected. Unfortunately, EASYRAD's GPR was fried at battery charging and collaboration with its developer was discontinued.

Own project: Easy Ground Penetrating Radar (2015 – 2018)
Complex co-design: PCB, mechanical and casing



PCBs, panels, labels and other staff development (Eagle, AutoCAD and Corel Draw) and production (incl. laser/water cutting)

The complex co-design of all electrical and mechanical components was carefully executed with particular attention to functionality and third-party manufacturability in small quantities on an acceptable price. In design stage was used Eagle, AutoCAD and Corel Draw SW packages. At production were used modern technologies like plastic foil panels, laser, water cutting etc. Serious attention was also paid to the interior and exterior aesthetics without making any compromises with functionality.

Own project: Easy Ground Penetrating Radar (2015 – 2018)
Investigation of hardware reliability with long term high load functional testing methodology

CPU, PMU and battery power and temperature characteristics of A20-OLinuXino-LIME2-eMMC (HW rev. E) based embeddable system

General conditions:

- The board is mounted in 160x165x51.5 mm aluminum box with 15mm thick aluminum thermo-conductor (cooler);
- The cooler contacts to all chips via thin (less than 1.5 mm) thermal pads (6 W/mK) and to the box wall via ARCTIC MX-4 thermal compound (with carbon micro-particles);
- Measurement is done with RPI Monitor on Armbian 5.12 with U-Boot 2016.5 and Linux Kernel 4.5.5 under application test load for whole discharge / charge cycle.

Note: The higher voltage drop problem at the first test was solved with changing the power wires with proven quality 18 AWG once.

High load test conditions:

- Stress tests: (while true; do stress -c 1 -i 1 -t 600 done);
- Network performance test: iperf -s + (while true; do iperf -c GBIT_IF_IP done) on i7/Windows 7/VMWare 8/Ubuntu 14 LTS box
- Application test: node.js server (started from rc.local) + JS client running in Chrome browser on i7/Windows 7 box

Notes: stress and server side iperf test staff was started from ssh consoles from PuTTY on i7/Windows 7 box.
CPU total load (as reported by RPI Monitor): 4.4 – 5.9 (1 min averaged), 5.1 (15 min averaged)

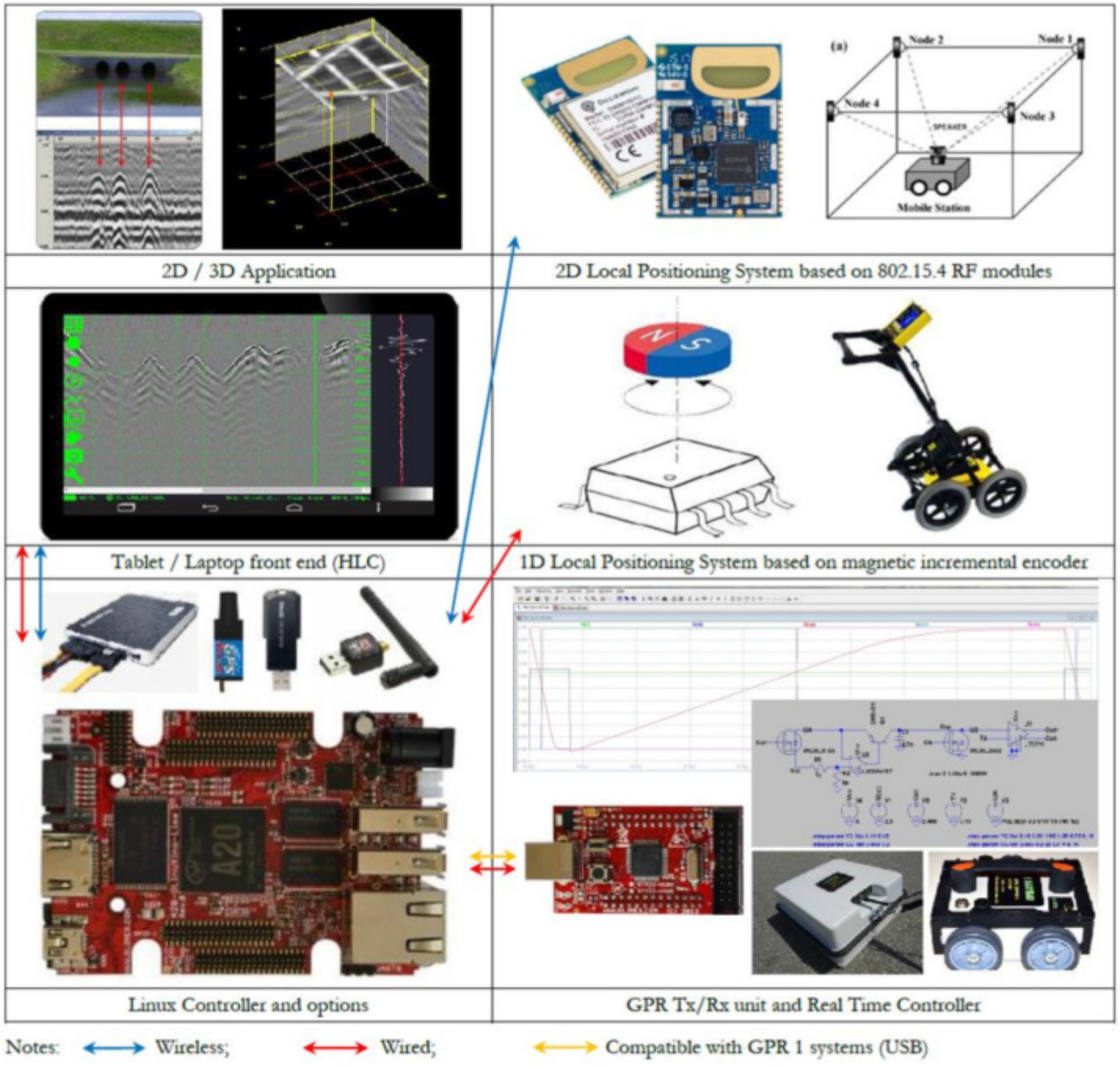
High temperature test conditions:

- All staff was put in a thermally isolated chamber (300 x 240 x 120 mm);
- Automotive 12 V / 21 W lamp and manually regulated power supply was used to heat the chamber air (dissipated power by device 2.5 W and 11.5 W additional);
- Computer 12V ventilator was used to homogenize the chamber air;
- First DS18B20 sensor and standalone digital thermometer ware used to measure the chamber air temperature;
- Second DS18B20 sensor was mounted on the aluminum thermo-conductor (cooler);
- PMU and CPU temperatures ware measured by armbianmonitor middleware.

First test – system in a box with cooling system	Second test – with modified power wires	Third test – with additional high load	Forth test – at 40°C environment temperature
• Discharge time: 10 hours at 2.5 W max • Charge time: 8.8 hours at 7.5 + 1.5 W (6910 Ah to battery) • Maximal temperatures at discharge cycle: 25°C PMU and 42°C CPU • Maximal temperatures at charge cycle: 37°C PMU and 42°C CPU • Up to 0.8V at 1.8A voltage drop was noticed in input power chain with side effects	• Discharge time: 10.5 hours at 2.5 W max • Charge time: 7.5 hours at 6 + 1.5 W (6740 Ah to battery) • Maximal temperatures at discharge cycle: 24°C PMU and 40°C CPU • Maximal temperatures at charge cycle: 35°C PMU and 42°C CPU • Up to 0.5V at 1.4A voltage drop was noticed in input power chain without side effects	• Discharge time: 6.8 hours at 3.6 W max • Charge time: 7.8 hours at 8.6/7.3 + 1.4/2.7 W (6170 Ah to battery) • Maximal temperatures at discharge cycle: 29°C PMU and 43°C CPU • Maximal temperatures at charge cycle: 44°C PMU and 45°C CPU • Up to 0.7V at 2A voltage drop was noticed in input power chain with side effects	• Discharge time: 10 hours at 2.5 W max • Charge time: 7.7 hours at 6.7 + 1.5 W (6740 Ah to battery) • Maximal temperatures at discharge cycle: 37°C PMU and 48°C CPU • Maximal temperatures at charge cycle: 48°C PMU and 49°C CPU • Up to 0.5V at 1.4A voltage drop was noticed in input power chain without side effects

Investigation of hardware reliability was done with long term high load functional testing methodology applied mainly to the Linux box based on A20-OLinuXino-LIME2-eMMC (HW rev. E). It was found problems with the heat transfer causing interior overheating and system crashes. That is why metal case was chosen and special passive cooling system was designed and tested. Massive heat-conductor was used to take away the heat from all hot chips on the board to the case. Thermal paste and a special thermal self-adhesive film, both with high quality, were used to improve the contact with the chips and the case.

Own project: Easy Ground Penetrating Radar (2015 – 2018)
Reapplied: Complex co-design and Long term high load functional testing methodology



It was started common project for development of flexible and contemporary GPR platform in cooperation with EASYRAD, developer and producer of series of such devices. It was planned that the mechanical design and analog components would remain as they were originally, but the digital ones had to be developed from scratch. Distributed architecture was chosen based on STM32F405 MCU/ChibiOS for the real time tasks, A20-OLinuXino-LIME2-eMMC/Linux for the server staff and tablet or laptop with any OS for the client staff. Software is planned to be written on C, JavaScript and HTML 5 and to be based on Secure Web Sockets, Https for both server and client staff. Single and two dimensional Local Positioning System (1D/2D LPS) and 2D/3D visualisation would also be a part of the future development. Production and sales prices can double at most.

How To projects (2015 – 2020)
Development, testing, documenting: Written documentation and references

Armbian customization how to

Armbian

Ubuntu/Debian images for ARM based single-board computers: http://www.armbian.com

Armbian main links

- Linux for ARM development boards
- Basic Documentation
- Advanced Documentation
- Armbian build tools on Git

Armbian Forum Links

- Armbian customization - main thread
- Add support for A20-Olinuxino-Lime2-eMMC
- Olimex Lime A20 Wont boot armbian
- Added Lime2 eMMC support
- Testers wanted: sunxi adjustments for RPi-Monitor
- [WiP] axp209 mainline sysfs interface

Olimex Forum Links

- A20-OLinuXino-LIME2 new HW Rev. E
- How to use eMMC on A20-OLinuXino-LIME2-eMMC
- Extremely slow Ethernet on A20-OLinuXino-LIME2
- Olinuxino A20 Lime 2 problems

Linux SUNXI User Group

- [PATCH 1/1] ARM: dts: sunxi: Add a olinuxino-lime2-emmc
- USB OTG on A20 Lime2 board does not work with mainline kernel
- [PATCH] musb: sunxi: Ignore VBus errors in host-only mode
- [PATCH 07/15] musb: Add support for the Allwinner sunxi musb controller

Our projects SVN links

- GPR Linux
- GPR Node JS
- GPR Web
- GPR Armbian

How to build Armbian (official notes)

Preparation

- x86/x64 machine running any OS; 4G ram, SSD, quad core (recommended),
- VirtualBox or similar virtualization software **(highly recommended)**,
- alternatively - Docker, systemd-nspawn or other containerization software,
- compilation environment is **highly recommended** to be Ubuntu Trusty 14.04 or Ubuntu Xenial 16.04 (other releases are **not officially supported**),
- installed basic system, OpenSSH and Samba (optional),
- superuser rights (configured `sudo` or root shell).

Execution

```
apt-get -y install git
git clone https://github.com/igorpecovnik/lib --depth 1
cp lib/compile.sh .
./compile.sh
```

376 lines (2015)

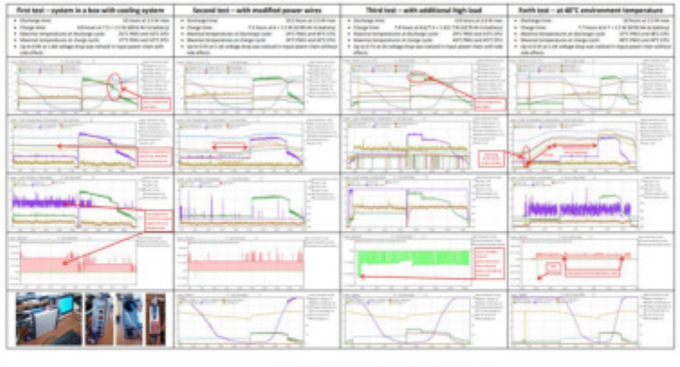
Armbian customization details

The target of customization described is to get production ready Armbian build for A20-Olinuxino-Lime2-eMMC board comprising with user defined requirements.

Armbian customization requirements

- Usage of main Armbian code base without modification
- Build process has to be non interactive and capable to be automated
- All customization stuff has to be:
  - delivered from own repository
  - applied mainly at the build process
  - only limited actions (preferably none) can be finished at the first run
- Final firmware has to be capable to be:
  - delivered to the user as a single file (as small as possible)
  - self sufficient and closed (not allowed later package updates)
  - updated on the field easily and with minimal user assistance
  - restored easily after any system crash
- The system has to:
  - function autonomously from the very beginning
  - be stopped and (re-)started with a single button touch
- User will interact with the system only via
  - the web interface and
  - a limited number of buttons and LEDs
- Additional user hardware has to be connected via available interfaces:
  - 2x USB Host and 1x USB OTG
  - 1x GBit native LAN
  - SATA and others bringing out GPIO connectors
  - Serial console will be used for development, production and maintenance only
  - HDMI and USB keyboard can be used in development process optionally as well
- Following HW will be attached to the system permanently:
  - Li-Po battery granting long life time and charged via DC-IN connector
  - eMMC containing fully functional and read only space copy of the system
  - SATA SSD containing system and user data
  - USB Wi-Fi Adapter working in "Access Point" mode
  - user buttons, LEDs and edge connectors
  - fully hermetic metal case with water proof edge connectors and controls
  - efficient cooling system taking off the heat via case walls
- Following HW will be attached to the system optionally or on demand:

CPU, PMU and battery power and temperature characteristics of A20-OLinuXino-LIME2-eMMC (HW rev. E) based embeddable system



763 lines (2015)

How to make MacBook Pro (15-inch, Late 2008) and iPad 2 useful in the year 2020

This blog is written to express my gratitude to all the people helped me to fulfill this initiative

Here one can find the steps for hardware upgrade of MacBook Pro (15-inch, Late 2008) and installation on it and iPad 2 the latest supported by Apple MacOS X v. 10.11.6 El Capitan, iOS v. 9.3.5 and MS Windows 10. All the steps made took more than a month reading, testing, more than 6 hours phone talks with Apple Support and as an exception tens of e-mails. More than 5 friends of mine were also involved in this tedious battle.

On the other hand because of my ignorance of contemporary Apple hardware and software some of the problems were definitely provoked by me. Some of the obvious facts written in this blog are also a consequence of the lack of knowledge and experience so be indulgent.

1. **Hardware upgrade**
   - The main hardware of MacBook Pro (15-inch, Late 2008) is:
     - Intel Mobile Core 2 Duo CPU P8600 @ 2.40GHz
     - RAM 4GB DDR3 @ 1066MHz and 250GB SATA I HDD
     - 256MB NVIDIA GeForce 9600M GT Graphics
     - 15.4-inch, 1440 x 900 pixels LCD with 32-bit color
   - Additional but helpful hardware of MacBook Pro (15-inch, Late 2008) is:
     - Gigabit Ethernet connector – preferable to Wi-Fi (better as speed and reliability)
     - Dual USB 2.0 and Firewire connectors – useful to connect external devices

1.1. Repairing of MacBook Pro (15-inch, Late 2008) and some workarounds

- One big trouble was that the battery fails and has to be changed
- At the same time failed capacitor on the motherboard makes a trouble bigger
- After changing the battery and repairing MacBook become ready for upgrade
- It is useful to know that MacBook Pro (15-inch, Late 2008) can work without battery
- One more problem was failed internal DVD device and has to be used external one connected via USB
- Unfamiliar language specific keyboard can also be a problem so it can be used additional one via USB
  - The problem at using unfamiliar keyboard is passwords entering when the chars are not shown at all
  - Some special characters can also be a problem to be entered even at visible fields

1.2. Changing memory from 4GB to 8GB

- It can be upgraded up to 8GB (there are some modules recommend)
  - https://eshop.macsales.com/guides/Mac_OS_X_Compatibility
- 1.2.1. Search and buy 2 modules 4GB DDR3, 1.5V, 1066 or 1333 MHz
- 1.2.2. Modules have to be the same type and from the same manufacturer
- 1.2.3. Modules can be from different manufacturers but with the same parameters
- 1.2.4. Test them with memtest software installation
- 1.2.5. Successfully used 4GB DDR3 1333MHz different modules:
  - Samsung, PC3-10700 (667 MHz), Part Number M471B5273CH0-CH9
  - Kingston, PC3-10700 (667 MHz), Part Number 9905428-051.A00LF
- 1.2.6. Do not install them before complete new software installation

1.3. Changing internal HDD with SSD

- It can be upgraded up to 4TB internal HDD or SSD (there are some devices recommend)
  - https://eshop.macsales.com/guides/Mac_OS_X_Compatibility
  - https://everymac.com/systems/apple/macbook_pro/macbook-pro-unibody-faq/macbook-pro-13-15-17-mid-2009-how-to-upgrade-hard-drive-ssd.html
  - https://www.ifixit.com/Answers/View/142274/What+is+the+maximum+size+hard+drive+I+can+put+in+my+Mid+2010+MB+Pro
  - https://www.howtogeek.com/348398/can-you-upgrade-the-hard-drive-or-ssd-in-your-mac/
  - https://www.macworld.com/article/3252169/ssd-3-old-macbook-compatibility.html
- 1.3.1. Original HDD: Hitachi HTB-HTS543232L9SA0, 250GB, 5400RPM, 2.5" (0A57323); SATA I (150Mbps)
- 1.3.2. Buy cheapest 480GB SSD with option to return in case of incompatibility
- 1.3.3. Take in account that MacBook Pro (2.40 GHz, 15-inch, Late 2008) has SATA II controller
- 1.3.4. It could be incompatible with some SSDs especially expensive and fast once
- 1.3.5. There is indication that Adata, Crucial and Kingston SSDs may work
- 1.3.6. Successfully used 447GB Crucial CT480BX300SSD1 ATA Device (SATA /SSD)
- 1.3.7. Do not install it before preparation of the software installation

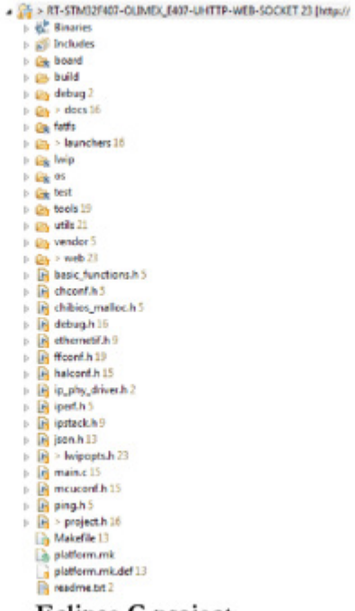
2. **Installation of MacBook Pro for dual boot of MacOS X and Windows 10**
   - There is no official Apple support for Windows 10 as Boot Camp and drivers for old MacBooks
   - Fortunately, all drivers for MacBook Pro (2.40 GHz, 15-inch, Late 2008) are available from Internet
     - Successfully updated after Windows installation by Snappy Driver Installer Origin
   - The only problem is multi touch drivers but there is an overcoming procedure (not tested)
     - https://discussions.apple.com/thread/1747021

405 lines (2020)

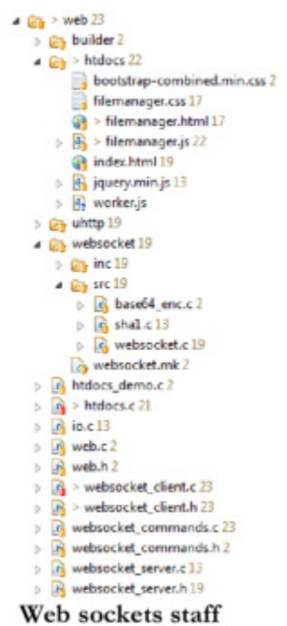
An own Linux distribution is planned for EGPR project so investigation about appropriate building platform was started. Armbian was chosen and evaluated. While studying customization details around A20-OLinuXino-LIME2-eMMC/Debian and adapting it to project requirements series of How-To notes were described. As a result, all notes and progress were documented and shared. Many aspects including results of long term high load functional testing were discussed in corresponding forums. Later on while MacBook Pro (15-inch, Late 2008) laptop and iPad-2 tablet were refurbished and improved all notes and progress were also described and shared as a How-To document.

Advanced embedded projects (2015 – 2018)

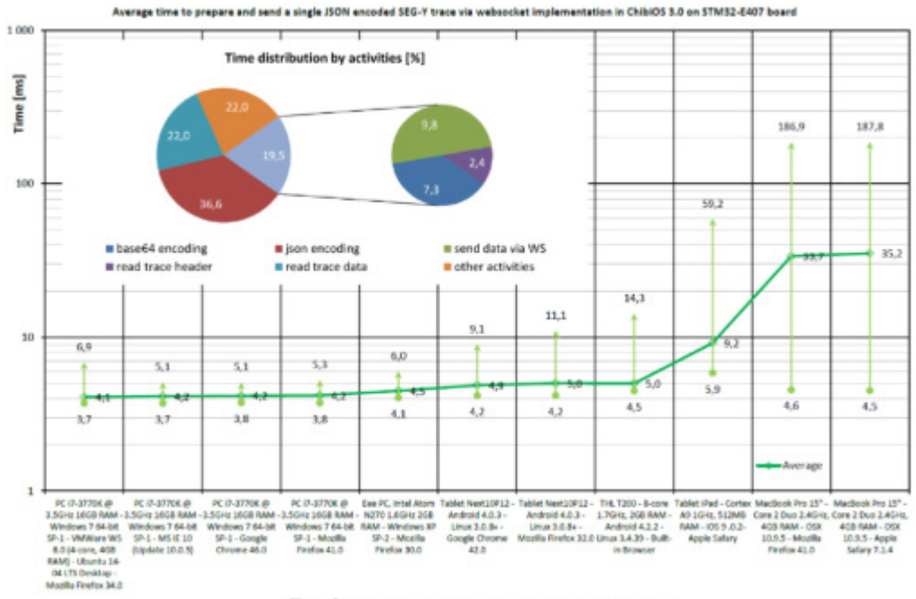
Olimex STM32-E407



Eclipse C project

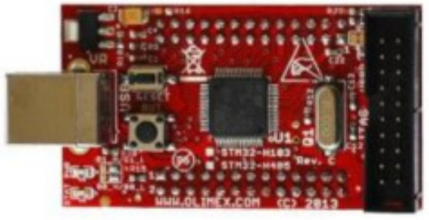


Web sockets staff

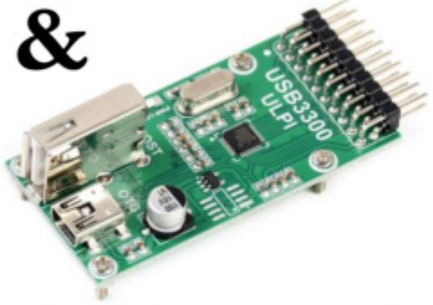


Performance measurement report

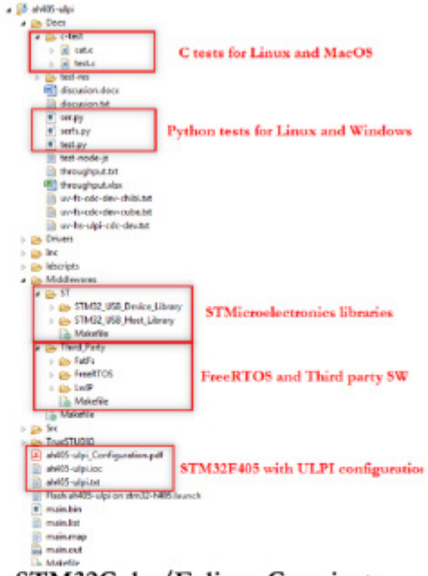
Advanced HTTP(S) and Web Sockets solution for embedded devices based on STM32F407 with Ethernet and ChibiOS



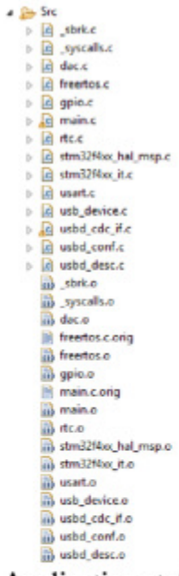
&



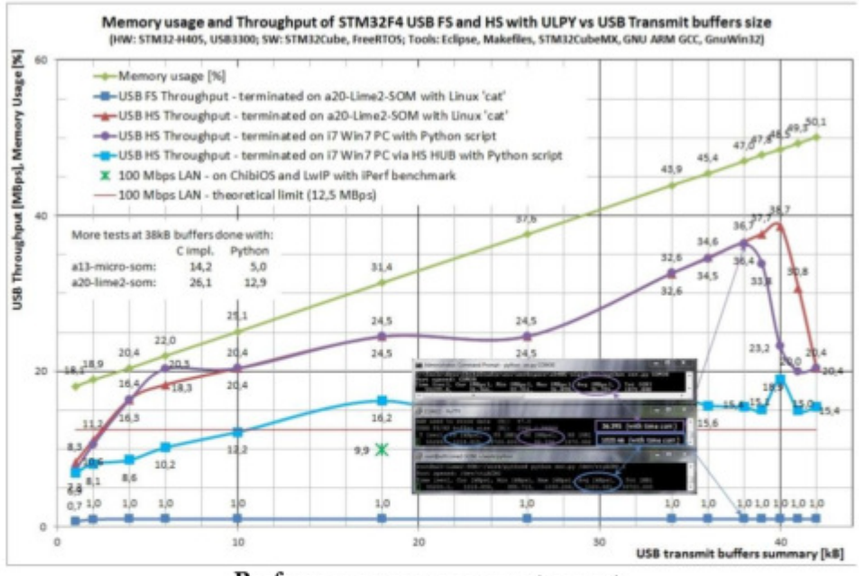
Olimex STM32-H405 & USB3300



STM32Cube/Eclipse C project



Application staff

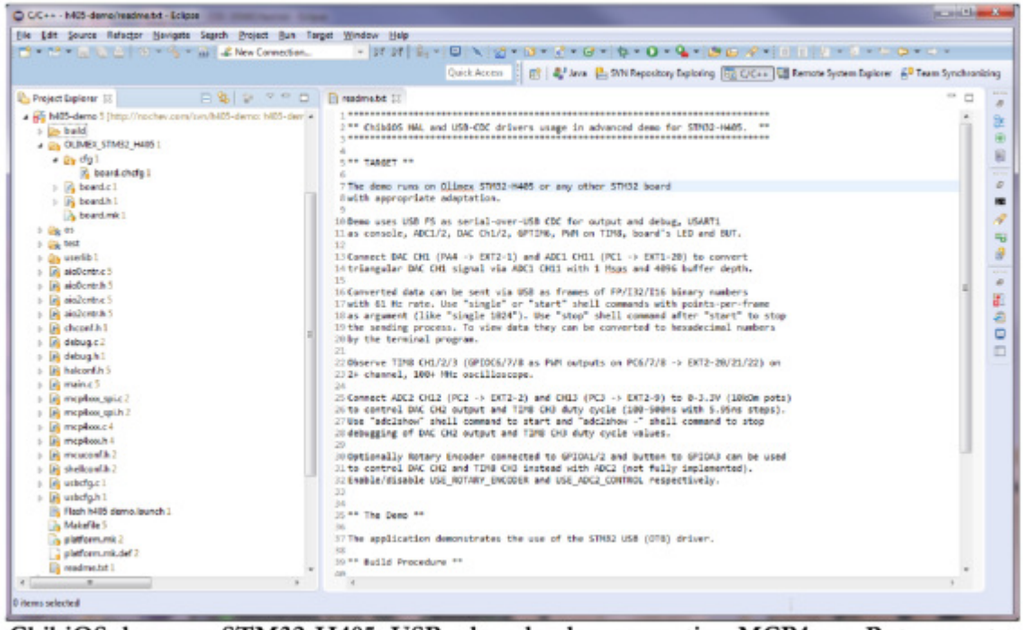


Performance measurement report

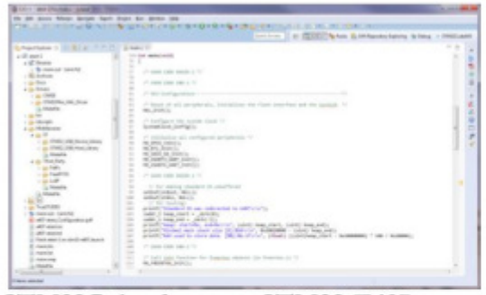
Advanced High Speed USB communication solution for embedded devices based on STM32F405, ULPI, USB3300 with STM32Cube and FreeRTOS

With respect to the forthcoming EGPR project HTTP(S) and Web Sockets services solution for embedded devices based on STM32F407 with Ethernet and ChibiOS was implemented and evaluated. Software implementation was very close to the real project requirements and series of measurements was made. Performance result was assessed carefully and the solution was rejected. For increasing of communication throughput USB high speed solution based on Olimex STM32-H405 & USB3300 ULPI chips are tested. The results were much better than at LAN scenario but other staff was limiting again so STM32F405/ChibiOS, A20-OLinuXino-LIME2-eMMC/Linux solution was chosen.

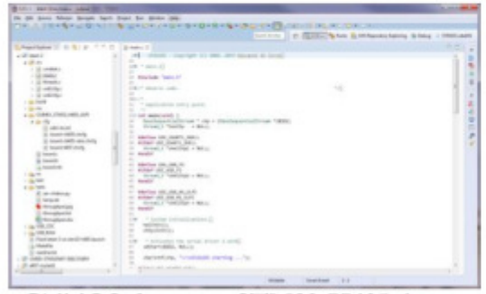
Advanced embedded projects – continue (2015 – 2018)



ChibiOS demo on STM32-H405: USB cdc, adc, dac, pwm, tim, MCP4xxx, Rot. enc. etc.

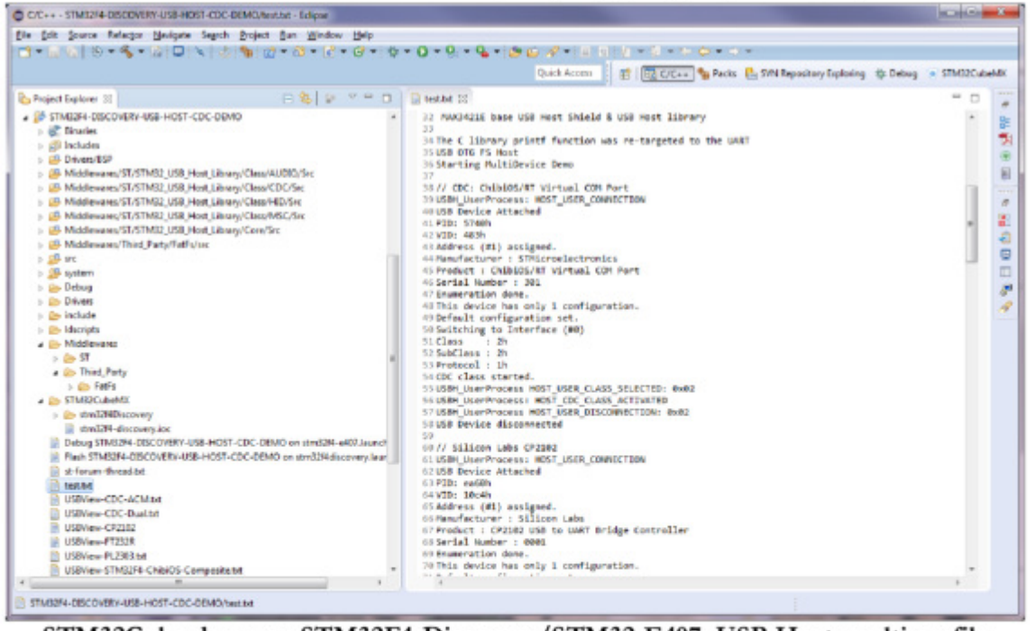


STM32Cube demo on STM32-E407: a-test

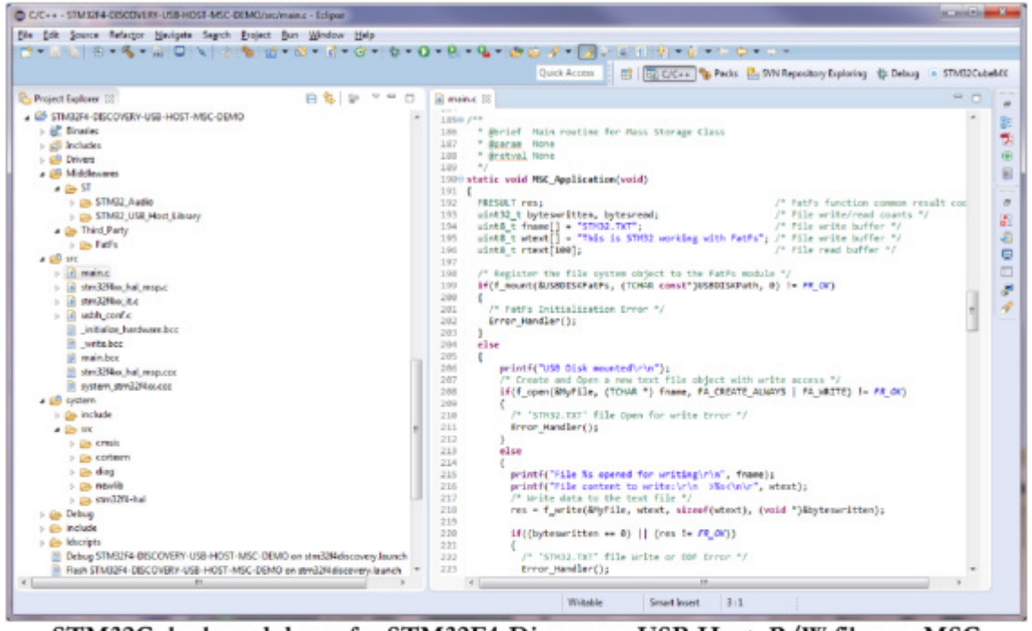


ChibiOS demo on STM32-H405: b-test

well tested and documented a/b-tests



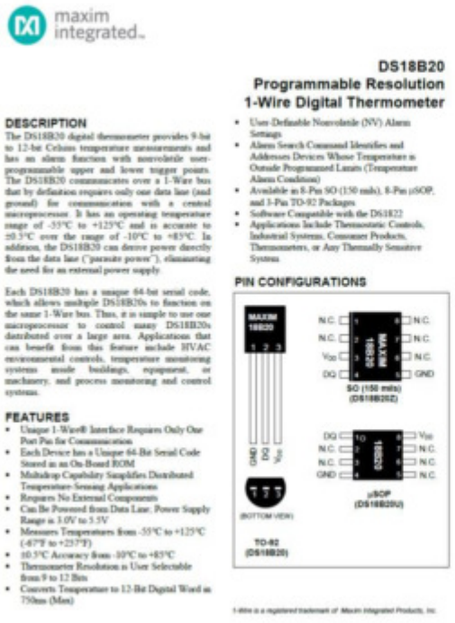
STM32Cube demo on STM32F4-Discovery/STM32-E407: USB Host, multi profile



STM32Cube based demo for STM32F4-Discovery: USB Host, R/W files on MSC

Series of advanced embedded projects were developed while studying and evaluating Cortex-M4 MCUs like STM32F405/7. Olimex STM32-H405, STM32-E407 and ST STM32F4-Discovery boards were used as hardware. Eclipse CDT, ChibiStudio, ChibiOS, STM32Cube, FreeRTOS and other mainly open source packages and libraries for Windows, Linux and MacOS were used as software. Big number of open source projects and demos were also used to build and evaluate really advanced embedded projects. All they were developed, discussed and shared with other people.

Helpful software projects (2015 – 2017)



maxim integrated

DS18B20
Programmable Resolution 1-Wire Digital Thermometer

DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire® bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line. Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS

SO (150 mils) (DS18B20Z)

µSOP (DS18B20U)

TO-92 (DS18B20)

(BOTTOM VIEW)

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Dalas 1-Wire Digital Thermometer

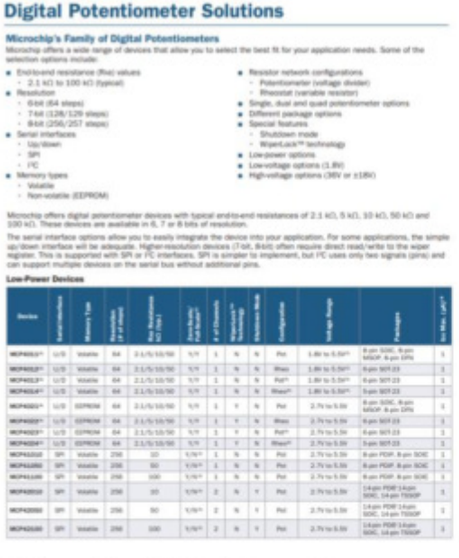
```
#ifndef OneWire_h
#define OneWire_h

#include <inttypes.h>
#include "Print.h"

#define FALSE 0
#define TRUE  1

class OneWire {
  private:
    uint8_t pin_DQ;
    // global search state
    unsigned char ROM_NO[8];
    uint8_t last_discrepancy;
    uint8_t done;

    uint8_t readScratchpad(unsigned char *deviceID, unsigned char *scratchpad);
    uint8_t writeScratchpad(unsigned char *deviceID, unsigned char *scratchpad);
    uint8_t copyScratchpad(unsigned char *deviceID);
    uint8_t getResolutionBits(unsigned char configByte);
    void writeByte(unsigned char byte);
    void writeBit(uint8_t bit);
    unsigned char readByte();
    uint8_t readBit();
    int touchByte(int byte);
    void block(unsigned char *data, int data_len);
    void select(unsigned char *deviceID);
    int crc8(unsigned char *deviceID, int length);
    int reset(void);
  public:
    OneWire(uint8_t pin);
    void resetSearch();
    uint8_t search(unsigned char *deviceID);
    uint8_t readPower(unsigned char *deviceID);
    uint8_t readPower();
    uint8_t broadcastConvertTemperature();
    uint8_t convertTemperature(unsigned char *deviceID);
    uint8_t convertTemperature();
    uint8_t setResolution(unsigned char *deviceID, uint8_t resolution);
    uint8_t setPermanentResolution(unsigned char *deviceID, uint8_t resolution);
    uint8_t getResolution(unsigned char *deviceID);
    float readTemperature(unsigned char *deviceID);
    float readTemperature();
};
#endif
```

```
#include "OneWire.h"

OneWire::OneWire(uint8_t pin) {
  pin_DQ = pin;
  resetSearch();
}

/*****************************************************************************/
/* Reset a OneWire Search                                                    */
/*****************************************************************************/
void OneWire::resetSearch() {
  uint8_t i;

  for(i = 0; i < 8; i++) {
    ROM_NO[i] = 0;
  }

  last_discrepancy = 0;
  done = FALSE;
}

/*****************************************************************************/
/* Issue a OneWire Search                                                    */
/*****************************************************************************/
uint8_t OneWire::search(unsigned char *deviceID) {
  /*
  The master begins initialization sequence by issuing a Reset Pulse.
  */
  uint8_t i, bit, complement_bit, byte_number, bit_index, discrepancy_marker;
  unsigned char byte_mask;

  if(done) {
    return FALSE;
  }

  // Perform Master Reset of OneWire Bus
  if(reset() == FALSE) {
    return FALSE; // device not found
  }

  // Issue the search command
  writeByte(0xF0);

  discrepancy_marker = 0;
  byte_number = 0;
  byte_mask = 1;
  bit_index = 1;
```

```
/*
  created on 10 June, 2014 by Mladen Nechev
*/

// include the library code:
#include <stdio.h>
#include <stdlib.h>
#include <LiquidCrystal.h>
#include <OneWire.h>

/*
Connection of LCD based on the Hitachi HD44780 (or a compatible) chipset:
 * LCD RS pin to -> D2   (PA0)  - D2
 * LCD Enable to -> D5   (PB6)  - D5
 * LCD D4 pin to -> D10  (PA4)  - D4
 * LCD D5 pin to -> D11  (PA7)  - D5
 * LCD D6 pin to -> D8   (PA15) - D6
 * LCD D7 pin to -> D9   (PB7)  - D7
 * LCD R/W pin to ground
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)
 */
//               (RS, En, D4, D5, D6, D7)
LiquidCrystal lcd(2, 5, 10, 11, 8, 9); // 8 x 2
OneWire oneWire(12); // pinDQ 12

#define BLUE_LED 7

struct oneWireDevice {
  unsigned char deviceID[8];
  int resolution;
  int mode;
  struct oneWireDevice *next;
};
struct oneWireDevice *root = NULL;

void setup() {
  int i;
  unsigned char deviceID[8];
  struct oneWireDevice *temp;
  struct oneWireDevice *current = NULL;

  // initialize the LED as an output:
  pinMode(BLUE_LED, OUTPUT);
  pinMode(BOARD_LED_PIN, OUTPUT);
  pinMode(BOARD_BUTTON_PIN, INPUT);
```

C++ Library

Arduino example



Digital Potentiometer Solutions

Microchip's Family of Digital Potentiometers

Microchip offers a wide range of devices that allow you to select the best fit for your application needs. Some of the selection options include:

- End-to-end resistance (Rwb) values
  - 2.1 kΩ to 100 kΩ (typical)
- Resolution
  - 6-bit (64 steps)
  - 7-bit (128/129 steps)
  - 8-bit (256/257 steps)
- Serial interfaces
  - Up/down
  - SPI
  - I²C
- Memory types
  - Volatile
  - Non-volatile (EEPROM)
- Resistor network configurations
  - Potentiometer (voltage divider)
  - Rheostat (variable resistor)
- Single, dual and quad potentiometer options
- Different package options
- Special features
  - Shutdown mode
  - WiperLock™ technology
- Low-power options
- Low-voltage options (1.8V)
- High-voltage options (36V or ±18V)

Microchip offers digital potentiometer devices with typical end-to-end resistances of 2.1 kΩ, 5 kΩ, 10 kΩ, 50 kΩ and 100 kΩ. These devices are available in 6, 7 or 8 bits of resolution.

The serial interface options allow you to easily integrate the device into your application. For some applications, the simple up/down interface will be adequate. Higher-resolution devices (7-bit, 8-bit) often require direct read/write to the wiper register. This is supported with SPI or I²C interfaces. SPI is simpler to implement, but I²C uses only two signals (pins) and can support multiple devices on the serial bus without additional pins.

Low-Power Devices

Device	Interface	Memory Type	Resistance (# of steps)	Rwb Resistance (Ω) (Typ.)	Wiper Configuration	Package
MCP4011*	U/D	Volatile	64	2.1/5/10/50	1/1	8-pin SOIC, 8-pin MSOP, 8-pin DFN
MCP4012*	U/D	Volatile	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP4013*	U/D	Volatile	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP4014*	U/D	Volatile	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP4021*	U/D	EEPROM	64	2.1/5/10/50	1/1	8-pin SOIC, 8-pin MSOP, 8-pin DFN
MCP4022*	U/D	EEPROM	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP4023*	U/D	EEPROM	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP4024*	U/D	EEPROM	64	2.1/5/10/50	1/1	6-pin SOT-23
MCP41010	SPI	Volatile	256	10	1/0/1*	8-pin PDIP, 8-pin SOIC
MCP41050	SPI	Volatile	256	50	1/0/1*	8-pin PDIP, 8-pin SOIC
MCP41100	SPI	Volatile	256	100	1/0/1*	8-pin PDIP, 8-pin SOIC
MCP42010	SPI	Volatile	256	10	1/0/1*	14-pin PDIP, 14-pin SOIC, 14-pin TSSOP
MCP42050	SPI	Volatile	256	50	1/0/1*	14-pin PDIP, 14-pin SOIC, 14-pin TSSOP
MCP42100	SPI	Volatile	256	100	1/0/1*	14-pin PDIP, 14-pin SOIC, 14-pin TSSOP

Microchip Digital Potentiometers

```
/*
 * mcp4xxx_spi.h
 *
 *  Created on: 22.03.2018 г.
 *      Author: Mladen
 */

#ifndef MCP4XXX_SPI_H_
#define MCP4XXX_SPI_H_

#include "hal.h"

void mcp4xxxSpiInit(mcp4xxxDevice *dev);
void mcp4xxxSpiExchange(mcp4xxxDevice *dev, size_t n, const void *txbuf, void *rxbuf);
void mcp4xxxSleep(uint16_t milliseconds);

typedef struct mcp4xxxSpiConfig_struct {
  SPIDriver *driver;
  const SPIConfig *config;
  uint16_t ack;
  uint16_t mask;
  uint16_t mask;
} mcp4xxxSpiConfig;

#endif /* MCP4XXX_SPI_H_ */


/*
 * mcp4xxx_spi.c
 *
 *  Created on: 22.03.2018 Fi.
 *      Author: Mladen
 */

#include "mcp4xxx.h"
#include "mcp4xxx_spi.h"
#include "hal.h"

void mcp4xxxSpiInit(mcp4xxxDevice *dev) {
  mcp4xxxSpiConfig *spi = (mcp4xxxSpiConfig *) dev->userdata;
  ioportid_t port = spi->config->sspout;
  uint16_t cs = spi->config->sspad;

  /*
   * SPI I/O pins setup.
   */
  palSetPadMode(port, spi->sck, PAL_MODE_ALTERNATE(5)  |
```

```
/*
 * mcp4xxx.h
 *
 *  Created on: 17.03.2018 Fi.
 *      Author: Mladen
 */

#ifndef MCP4XXX_H_
#define MCP4XXX_H_

#include <stddef.h>
#include <stdint.h>

#define MCP4XXX_MAX_REGISTER_VALUE 0x01FF

#define MCP4XXX_WIPER_0 0
#define MCP4XXX_WIPER_1 1
#define MCP4XXX_MAX_WIPERS 2

#define MCP4XXX_ADDR_WIPER_0_VOLATILE 0x00 //0b00000000
#define MCP4XXX_ADDR_WIPER_1_VOLATILE 0x10 //0b00010000

#define MCP4XXX_ADDR_WIPER_0_NON_VOLATILE 0x20 //0b00100000
#define MCP4XXX_ADDR_WIPER_1_NON_VOLATILE 0x30 //0b00110000

/**
 * Actually CMD_READ mask is 0x3C but with multiplexed SDI/SDO pin
 * it is important to have bit 0 and bit 1 set to 1 to avoid issues with not correct 9-bit data
 */
#define MCP4XXX_CMD_READ 0x0F //0b00001111
#define MCP4XXX_CMD_WRITE 0x00 //0b00000000

#define MCP4XXX_T_CON_REGISTER 0x40 //0b01000000
#define MCP4XXX_STATUS_REGISTER 0x50 //0b01010000

#define MCP4XXX_DATA_REGISTER_0 0x60
#define MCP4XXX_DATA_REGISTER_1 0x70
#define MCP4XXX_DATA_REGISTER_2 0x80
#define MCP4XXX_DATA_REGISTER_3 0x90
#define MCP4XXX_DATA_REGISTER_4 0xA0
#define MCP4XXX_DATA_REGISTER_5 0xB0
#define MCP4XXX_DATA_REGISTER_6 0xC0
#define MCP4XXX_DATA_REGISTER_7 0xD0
#define MCP4XXX_DATA_REGISTER_8 0xE0
#define MCP4XXX_DATA_REGISTER_9 0xF0

#define MCP4XXX_MAX_DATA_REGISTERS 10
```

```
/*
 * mcp4xxx.c
 *
 *  Created on: 17.03.2018 Fi.
 *      Author: Mladen
 */

#include "mcp4xxx.h"

/*
 * SPI TX and RX buffers.
 */
static uint8_t txbuf[2];
static uint8_t rxbuf[2];

void mcp4xxxInit(mcp4xxxDevice *dev) {
  dev->spi->init(dev);
}

/**
 * Register functions
 */

uint8_t mcp4xxxWriteRegister(mcp4xxxDevice *dev, uint8_t address, uint16_t data, uint8_t isVolatile) {
  txbuf[0] = MCP4XXX_CMD_WRITE | address;
  txbuf[1] = data;

  // Calculate the 9-bit data value to send
  if(data > 255) {
    txbuf[0] |= 0x01;
  }

  dev->spi->exchange(dev, 2, txbuf, rxbuf);

  if(!isVolatile) {
    // EEPROM write cycles take 4ms each. So we block with delay(8); after any NV Writes
    dev->spi->sleep(5);
  }

  return 0x01 & (rxbuf[0] >> 1);
}

uint16_t mcp4xxxReadRegister(mcp4xxxDevice *dev, uint8_t address) {
  txbuf[0] = MCP4XXX_CMD_READ | address;
  txbuf[1] = 0xFF;

  dev->spi->exchange(dev, 2, txbuf, rxbuf);

  return MCP4XXX_MAX_REGISTER_VALUE & ((uint16_t) rxbuf[0] << 8 | (uint16_t) rxbuf[1]);
}
```

C Library (Tested in STM32F405 ChibiOS project)

Dallas 1-Wire Digital Thermometers and Microchip Digital Potentiometers were rated as very useful devices for many use cases. Therefore, extensive research was done to find good software libraries and projects supporting these devices. Unfortunately, the resources found did not satisfy our requirements, so we decided to make our own implementation. Of course, the found projects are used and we respect their authors. Both projects were shared and used later on in other projects we worked on.

Own project: Powerful wind generator and test suite (2014 – 2016)
Reapplied: Complex co-design and Long term high load functional testing methodology

Generator Construction

- NdFeB Permanent Magnets – supplier selection and delivery

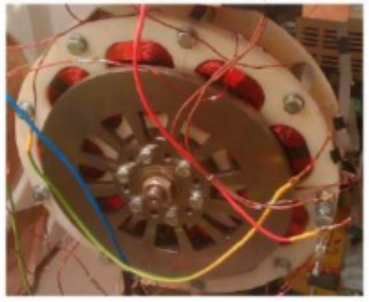
Dailymag Magnetic Technology (Ningbo) Limited is a Chinese leading manufacturer and exporter of permanent magnets etc.

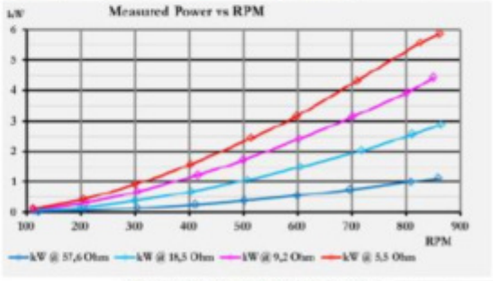
Wind Generator NdFeB Magnets 22.5 degree
8 inch OD x 4 inch ID x 0.5 inch thick
Wedge Segment Shape, Grade N35~N52
Nickel-Copper-Nickel triple layer coated

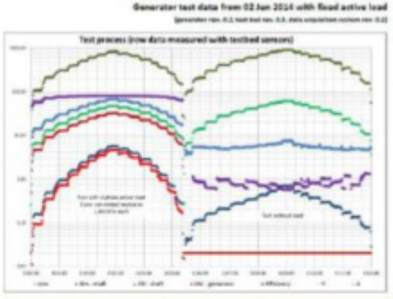
- Generator 2-nd rev., Test bed 3-rd rev., 3-phase load and data acquisition 1-st rev.

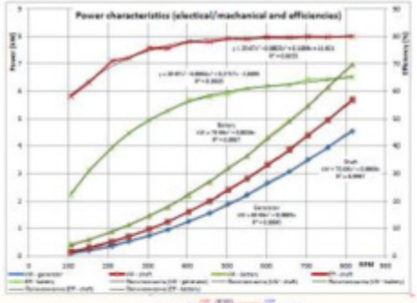
Generator Implementation

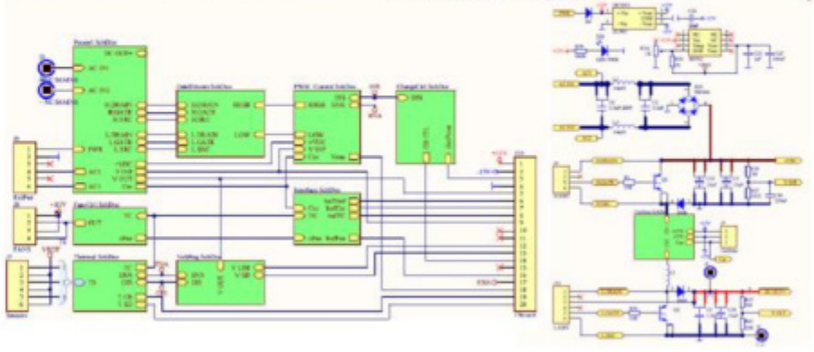
- Generator mounted on the test bed – (both first revision)





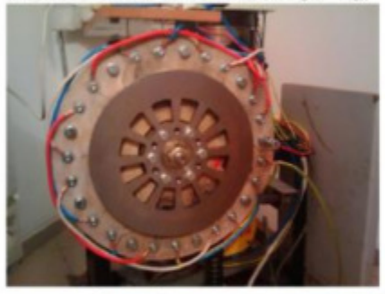


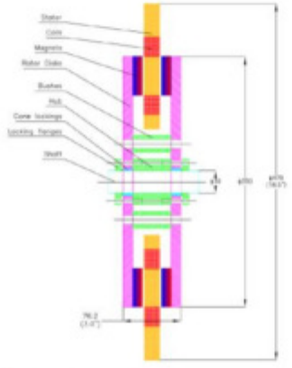




Generator Improvements

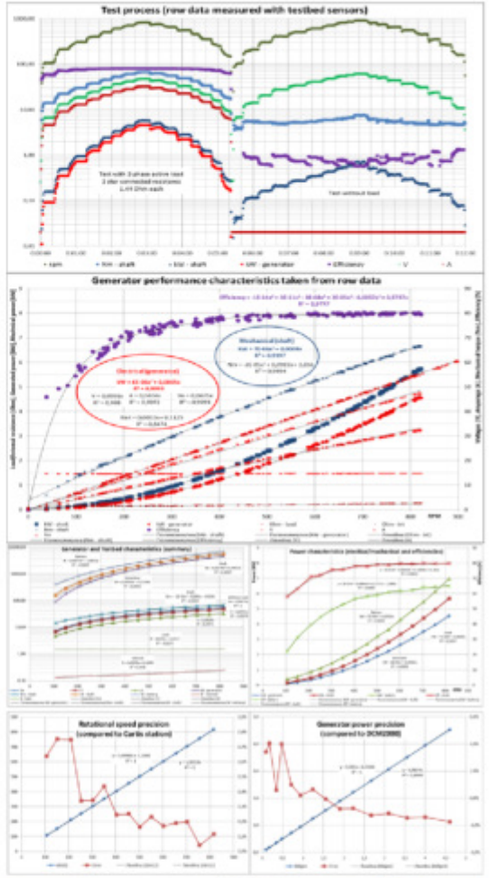
- Wind Generator – first revision with modified stator (molding technology)

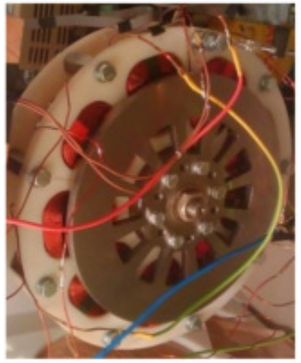


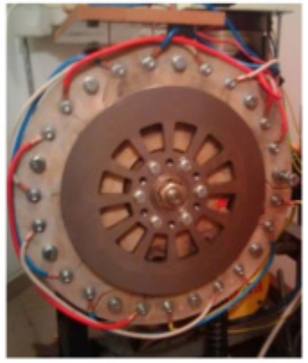


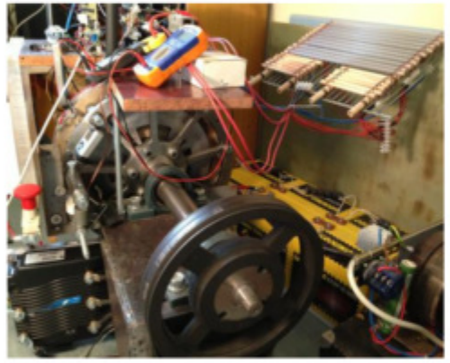
In addition to the described achievements, a prototype of a powerful (10 kW) DC/CD Up/Down converter was developed but not tested yet. Two generators were produced as prototypes. Two pilot installations were also made (high-speed spiral turbine at 1860 m altitude near the Belmeken dam and low-speed turbine with H-Blades at 64 m altitude near the town of Shabla). A couple of proposals for starting of generator production were also made but not accepted yet.

Own project: Powerful wind generator and test suite (2014 – 2016)
Long term high load functional testing methodology: generator and full-featured test suite



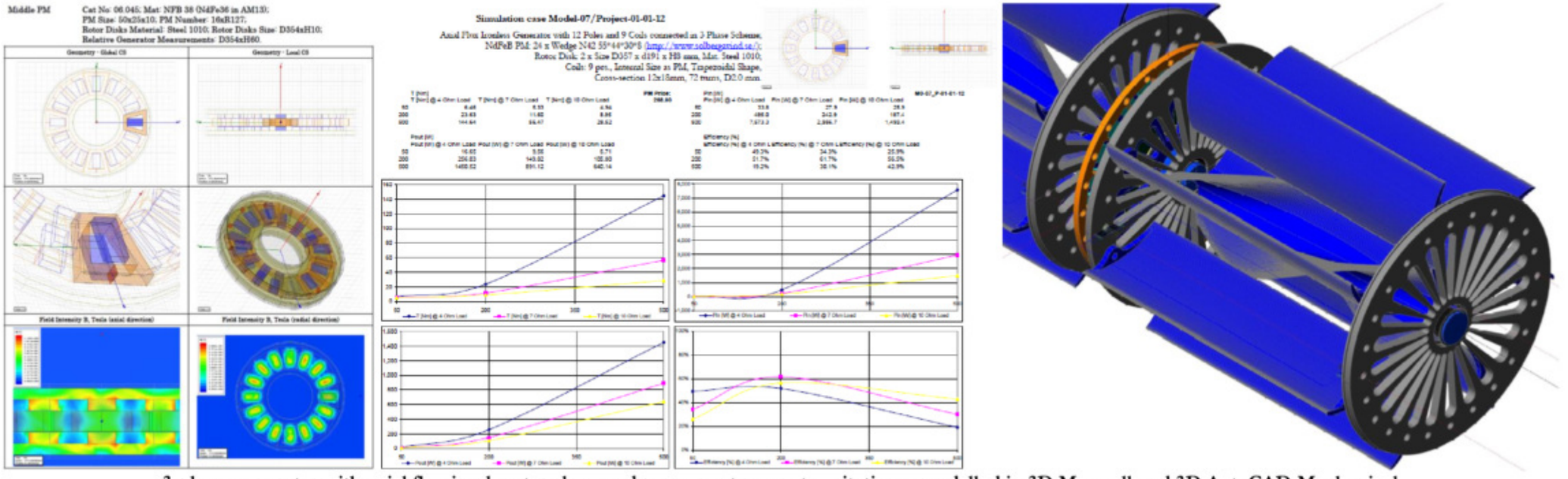




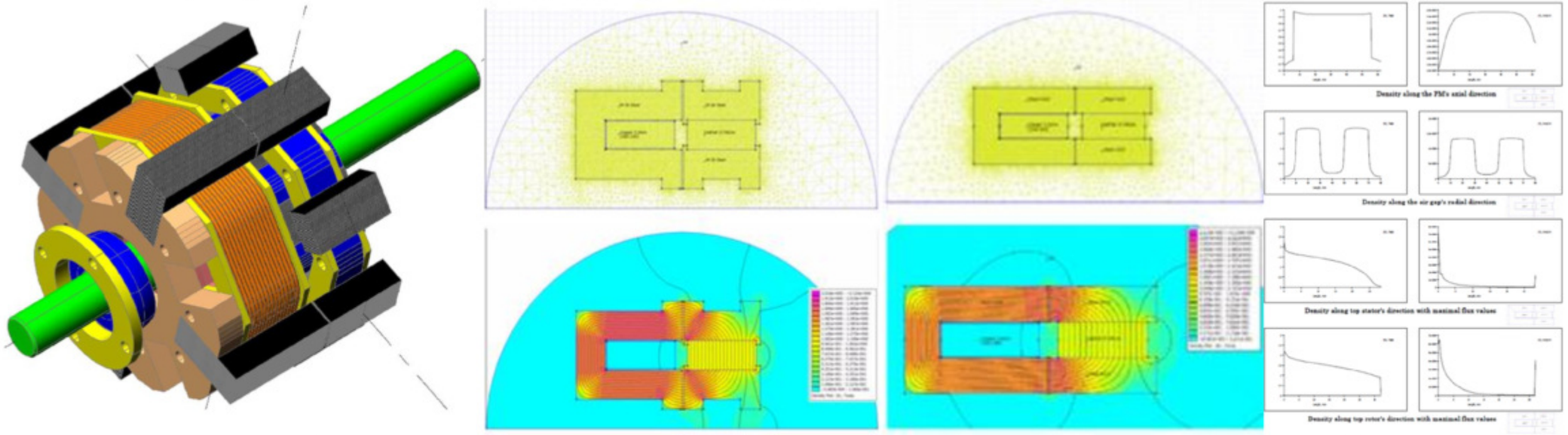


In addition to the selected generator, a test bed was also developed. Both the generator and the test bed underwent a series of modifications and even a complete redesign, but in the end a maximum power more than 5 kW was reached. Some of the successes include two variants each of the stator (melted and milled), the rotor (with massive and lightweight hub) and the coils (60 and 120 turns). The test bed in its second version was redesigned from scratch, and the drive went from a DC motor to a high-frequency AC one controlled by a special controller and powered by storage batteries. Three variants of the load resistor (DC with manual and electronic control, as well as three-phase AC) were also developed. Several variants of the electronic data collection system and off-line data processing methodology were developed and used as well.

Own project: Powerful wind generator and test suite (2014 – 2015)
Complex co-design: 2D/3D dynamical electromagnetic and 3D mechanical modelling



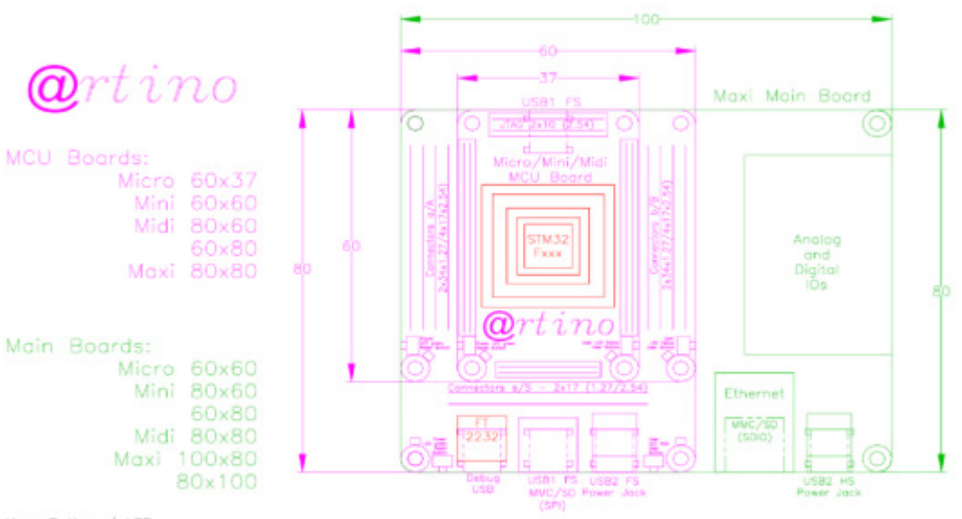
3-phase generator with axial flux ironless topology and permanent magnet excitation – modelled in 3D Maxwell and 3D AutoCAD Mechanical

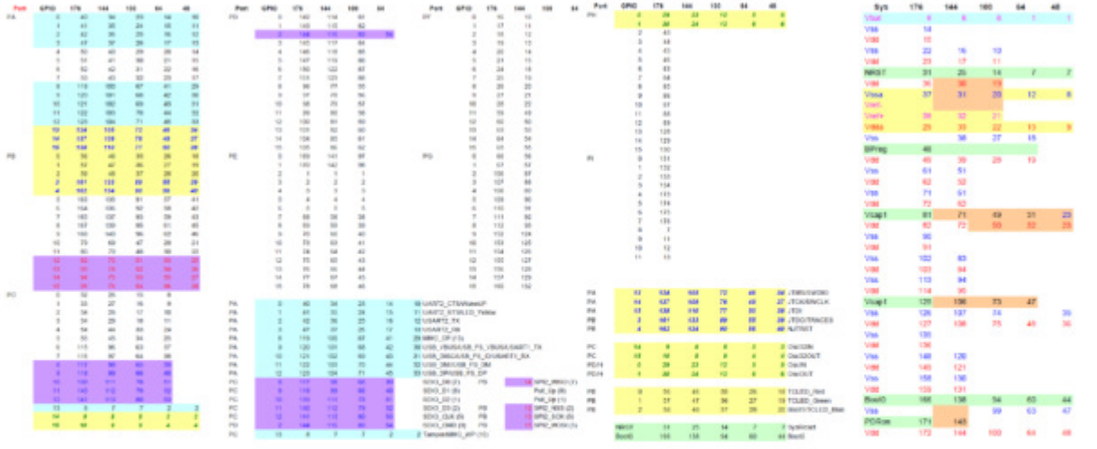


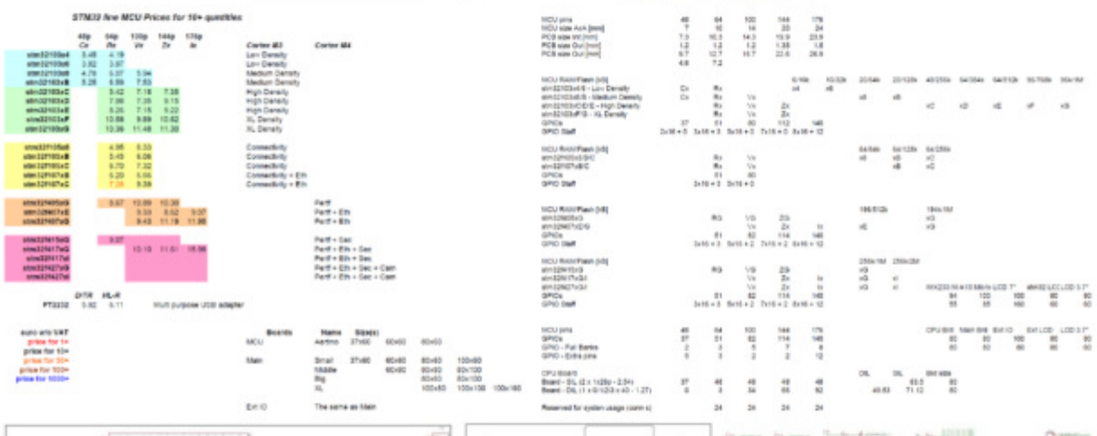
3-phase generator with concentrating transverse flux topology and permanent magnet excitation – modelled in 2D FEMM and 3D AutoCAD Mechanical

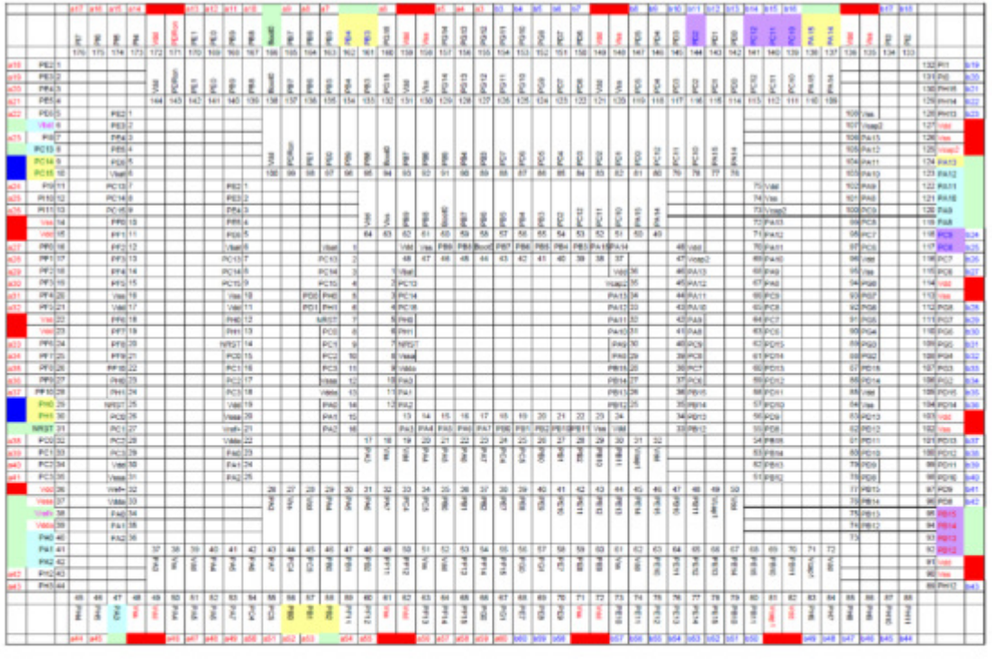
An own project for development, production and installation of small wind turbines was stared. First idea was to develop vertical spiral turbine and 3-phase generator with concentrating transverse flux topology and permanent magnet excitation. Later on, because of technological consideration 3-phase generator with axial flux ironless topology and permanent magnet excitation was chosen. All ideas were subjected to careful analysis and modelling both electro dynamically and mechanically. Final solution for the generator was to use strong NdFeB wedge permanent magnets for reaching 5 kW maximal power. For the turbine the estimation was to use static wings for increasing the wind power.

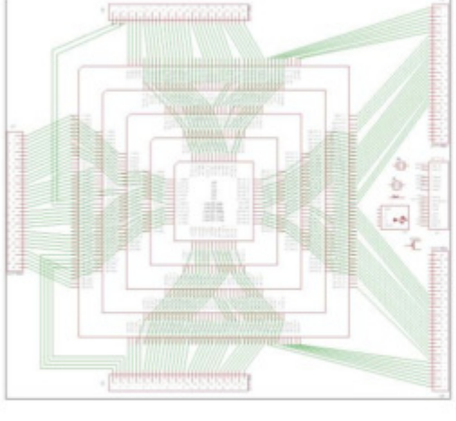
Own project: Arduino like staff based on STM32F1/4 and ChibiOS/STM32Cube (2014 – 2015)

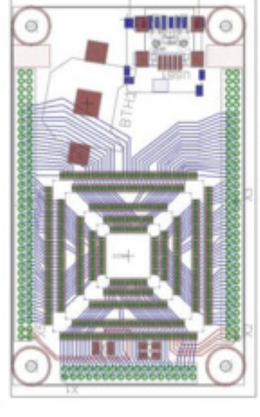


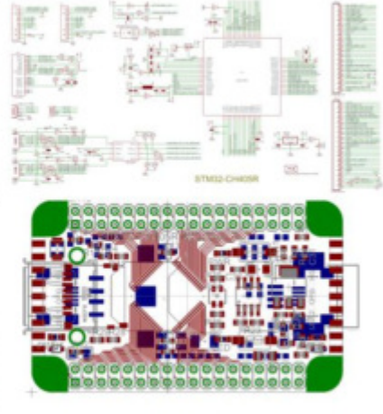






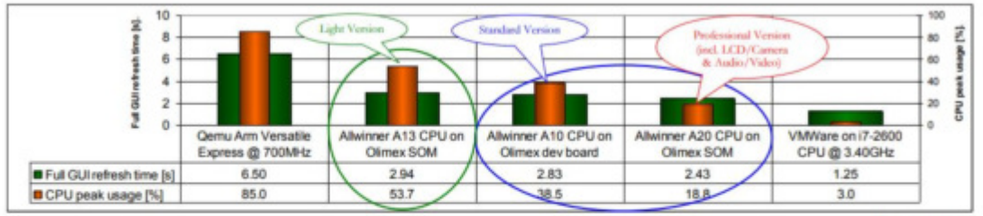






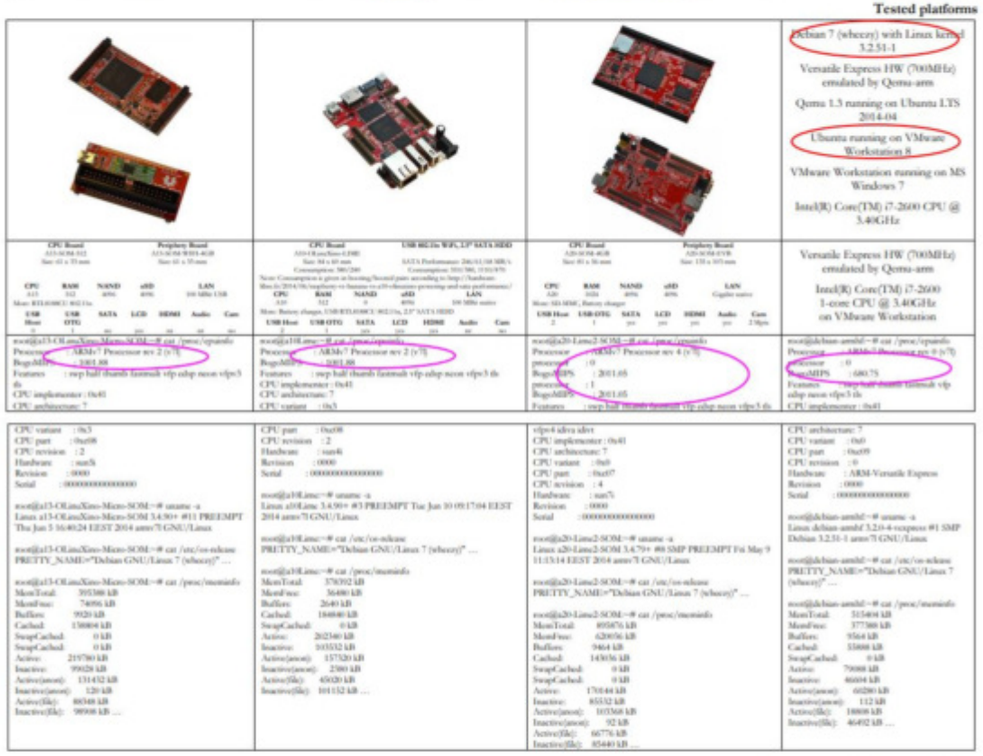
Arduino is very popular platform but it is suitable for small amateur projects mainly in the field of education. Unfortunately, attempts to extent it for professional and production use did not gain momentum. It was started a project for creating of easy to use professional platform based on Cortex-M3/4 MCUs with complete usage of their potential. The first idea to develop a single board for almost all chips did not prove practical. The other idea for a board as simple as STM32-H405 but with more connectivity (STM32-CH405R) was much more feasible especially combined with USB ULPI as hardware and ChibiOS or STM32Cube as software. Unfortunately, the project was not brought to a successful end.

Advanced solution proposal: NFC and Linux based Internet enabled Access Control System (2013 – 2014)

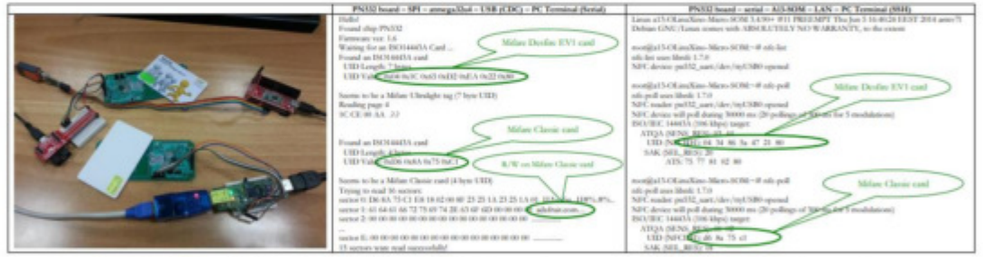


	Qemu Arm Versatile Express @ 700MHz	Allwinner A13 CPU on Olimex SOM	Allwinner A10 CPU on Olimex dev board	Allwinner A20 CPU on Olimex SOM	VMWare on i7-2600 CPU @ 3.40GHz
Full GUI refresh time [s]	6.50	2.94	2.83	2.43	1.25
CPU peak usage [%]	85.0	53.7	38.5	18.8	3.0

Tested platforms



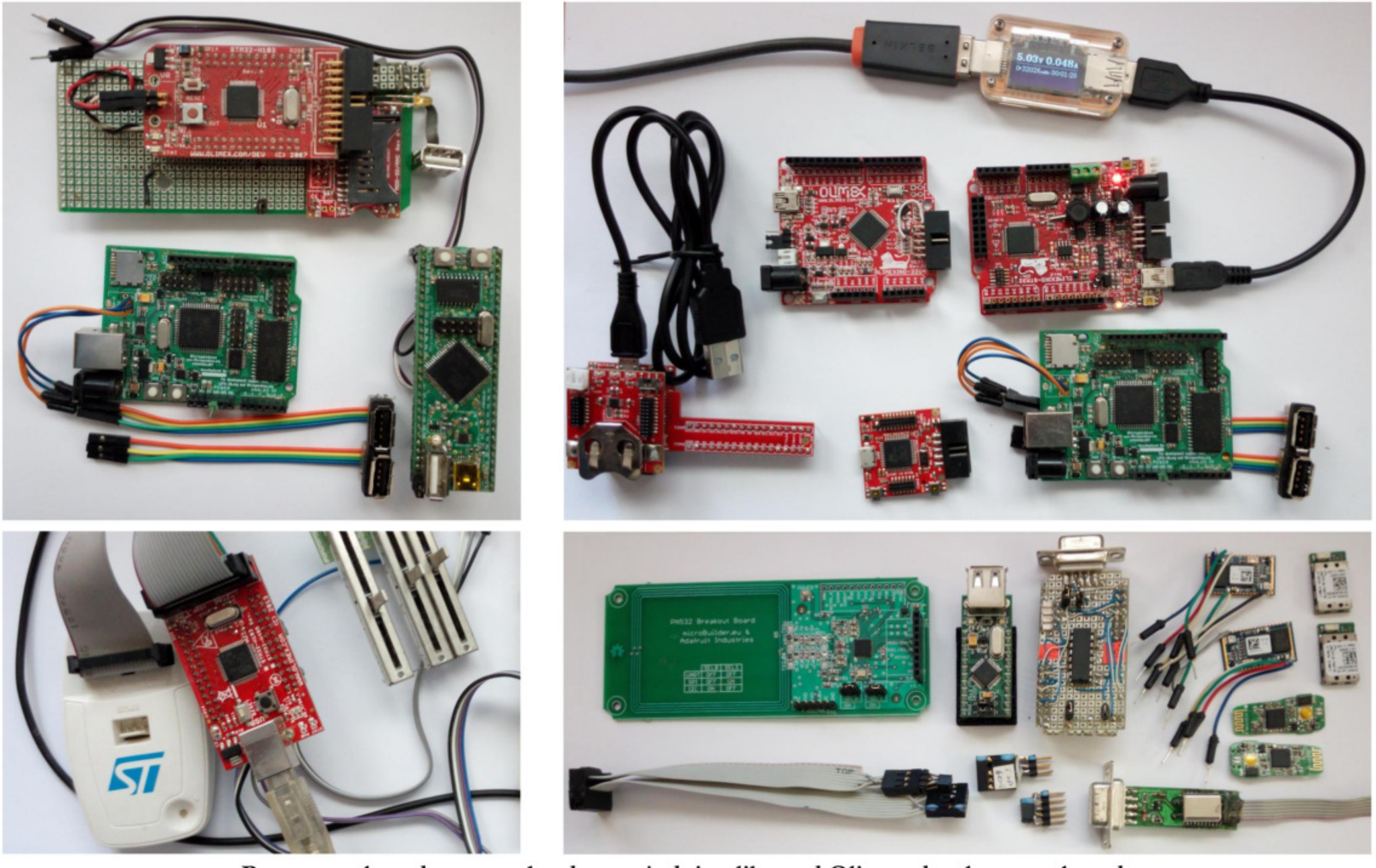
Preliminary tests with PN532 board



Embedded Linux with OSGI services and NFC card reader

Open source staff, application interface and performance tests

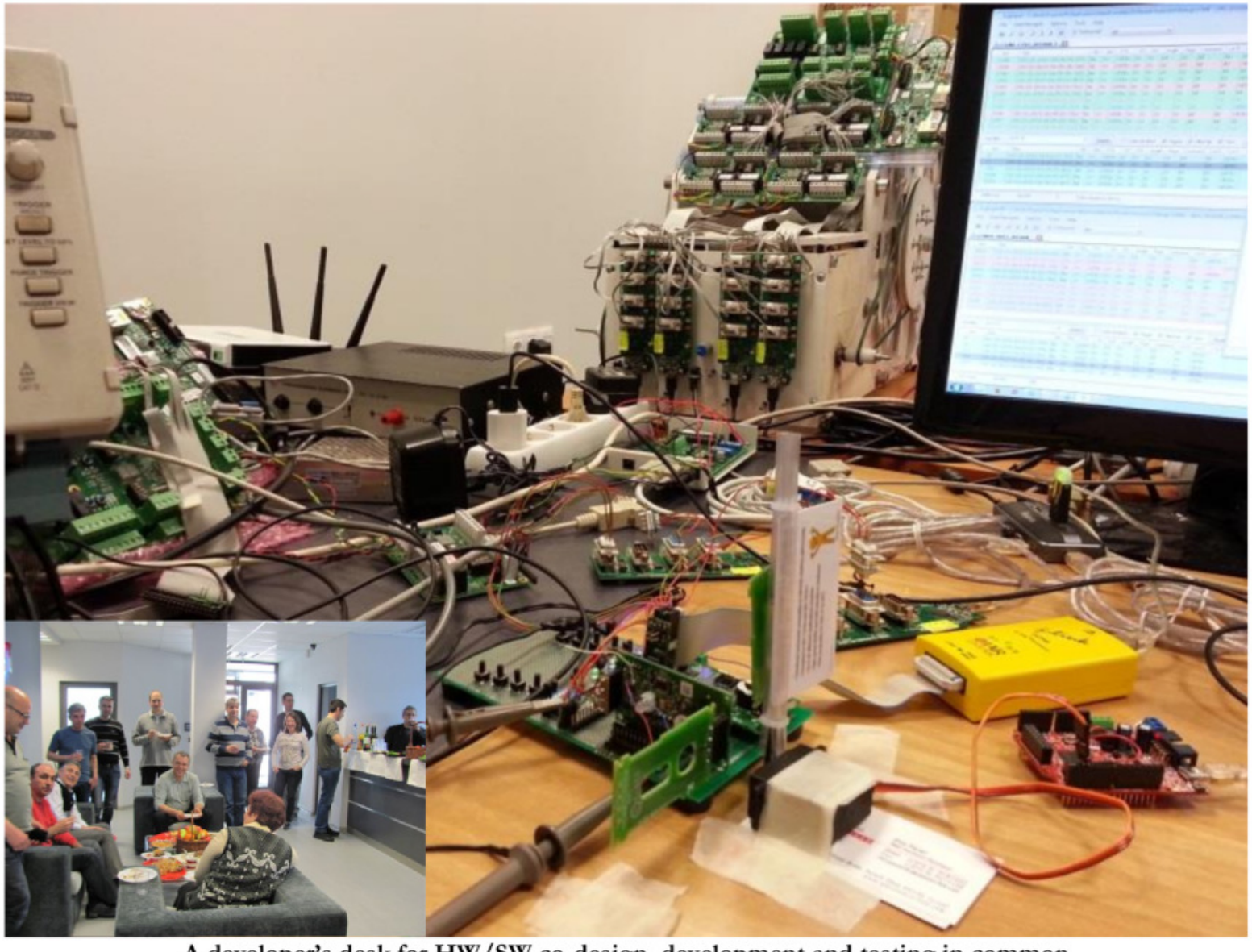
As a result of gained experience in the field of Internet enabled access control systems a proposal for migration to a new architecture was suggested. The main idea is to use NFC and Linux as a base for a new generation of intelligent card readers building ed hoc distributed system. Proposed solution was feasible because of great decreasing of the prices and the consumption of the Linux capable SOCs and can give extremely high level of flexibility and scalability avoiding power hungry server staff and intermediate infrastructure. Unfortunately, proposed solution did not gain popularity because security companies are very conservative and not opened for technological novelties especially open source based.

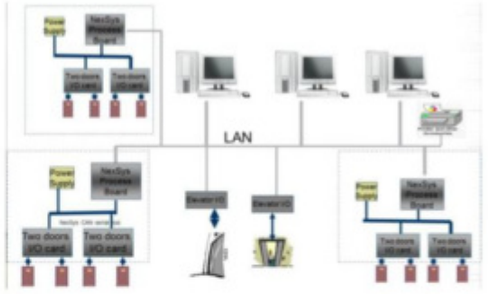


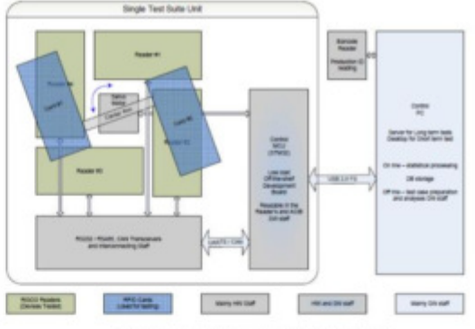
Prototypes based on open hardware, Arduino like and Olimex development boards

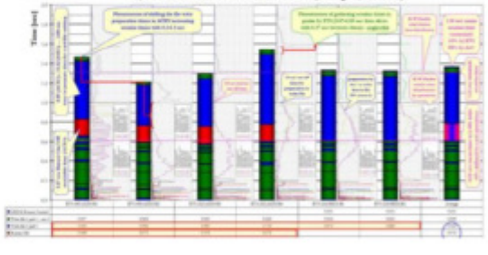
A number of MCU based development and open hardware boards were explored and evaluated for usage in various projects. Some were part of popular Arduino ecosystem, but others were for professional use like header and development boards from Olimex and other companies. Some prototypes were also made. Besides the MCU staff, many Linux capable SOCs and corresponding boards were also part of the devices used, including in combination with MCUs for splitting the real time and the processing hungry tasks.

RISCO: RFID card reader for Internet enabled Access Control System (2011 – 2014)
Reapplied: Long term high load functional testing methodology









A developer's desk for HW/SW co-design, development and testing in common

Inheriting the project for RFID card reader for Internet enabled access control system in a rather initial phase and without support especially in the area of testing, a new and different methodology was introduced combining the development and the testing in a common process. Long term high load functional testing methodology was accepted as a base and a special test suit including real infrastructure and complete server staff was developed, implemented and continuously used for massive testing of devices in real use case. This approach allowed all errors in the hardware and software, including those in the supporting infrastructure and server staff, to be immediately detected and fixed in the development process. Other big success was recognition by the firmware of different production variants (RS232, RS485 CAN) and ability to adapt silently.

How To projects (2011 – 2016)
Development, documenting, testing: Written documentation, examples and references

Building and setup Ångström distribution from scratch

Host platform: Ubuntu Desktop 10.10

Target platform: Olimex CS-E9302 Development board

1. Overview

Ångström was started by a small group of people who worked on the OpenEmbedded, OpenZaurus and OpenSimpad projects to unify their effort to make a stable and userfriendly distribution for embedded devices like handhelds, set top boxes and network-attached storage devices and more. All Ångström binaries are built using OpenEmbedded.

OpenEmbedded offers a best-in-class cross-compile environment. It allows developers to create a complete Linux Distribution for embedded systems. Some of the OpenEmbedded advantages include:

- support for many hardware architectures
- multiple releases for those architectures
- tools for speeding up the process of recreating the base after changes have been made
- easy to customize
- runs on any Linux distribution
- cross-compiles 1000's of packages including GTK+, Qt, the X Windows system, Mono, Java, and about anything else you might ever need

Bitbake handles the parsing and execution of the data files. The data itself is of various types; recipes which give details about particular pieces of software, class data which is an abstraction of common build information (e.g. how to build a Linux kernel) and configuration data for machines, policy decisions, etc., which acts as a glue and binds everything together. Bitbake knows how to combine multiple data sources together, each data source being referred to as a layer. Bitbake is responsible for parsing the metadata, generating a list of tasks from it and then executing them. The most common usage is **bitbake packagename** where packagename is the name of the package you wish to build (from now on called the target). This often equates to the first part of a .bb filename, so to run the simple-package_1.2.3.bb file, you might type **bitbake simple-package**. Several different versions of **simple-package** might exist and bitbake will choose the one selected by the distribution configuration. Bitbake will also try to execute any dependent tasks first so before building **simple-package** it would build a cross compiler and glibc if not already built.

The Metadata (recipes) are .bb files that are usually referred to as 'recipes'. In general, a recipe contains information about a single piece of software such as where to download the source, any patches that are needed, any special configuration options, how to compile the source files and how to package the compiled output.

Class (.bbclass) files contain information which is useful to share between metadata files. An example is the autotools class which contains the common settings that any application using autotools would use.

The configuration (.conf) files define various configuration variables which govern what Poky does. These are split into several areas, such as machine configuration options, distribution configuration options, compiler tuning options, general common configuration and user configuration (local.conf).

54 pages (2011)

How to install ChibiStrudio and use ChibiOS on Windows machine

At the date (Mar 27, 2015) of writing following ChibiOS and ChibiStudio (http://chibios.org/) versions are available for download from (http://sourceforge.net/projects/chibios/) and SVN (http://svn.code.sf.net/p/chibios/svn/trunk) and Git (https://github.com/ChibiOS/ChibiOS-RT) repositories:

- Stable ChibiOS/RT ver. 2.6.7 (http://sourceforge.net/projects/chibios/files/ChibiOS_RT%20stable/Version%202.6.7/);
- Development ChibiOS RT/NIL ver. 3.0 (http://svn.code.sf.net/p/chibios/svn/trunk);
- Contributor's stuff for ChibiOS RT/NIL ver. 3.0 (https://github.com/ChibiOS/ChibiOS-Contrib);
- ChibiStudio Preview 9 (http://sourceforge.net/projects/chibios/files/ChibiStudio/ChibiStudio_Preview9.7z/download).

The easiest way to start is to install ChibiStudio Preview 9 containing both ChibiOS/RT ver. 2.6.7 and ver. 3.0dev versions. Unfortunately, included Eclipse version Luna is for 32-bit Windows and will not work on a 64-bit host. To overcome this problem Eclipse IDE for C/C++ developers for Windows 64-bit has to be installed. This is tested on Windows 7/8 and it is applicable for newer Eclipse versions as well. Installation is also tested on VMWare Workstation with Windows 8 guest and work fine but there are problems with FW uploading via ST-LINK/V2 Utility. There is also problem to connect USB removable devices (Virtual com ports for example) when connected on some USB 3.0 ports of the host machine. There is not possible to move USB device to the VM when it is wrong enumerated. That is why installation on real Windows machine is recommended.

Description of main procedures required for successful startup:

1. **Download ChibiStudio last version and unzip it in C:\ChibiStudio\;**
2. **Install the drivers for JTAG/SWD adapter (ST LINK/V2) including STM32 ST-LINK Utility;**
3. **It is good idea (for make from command prompt) to add to the Windows PATH variable following folders:**
   a. C:\ChibiStudio\tools\gnutools\bin and
   b. C:\ChibiStudio\tools\GNU Tools ARM Embedded\4.7 2014q2\bin;
4. **Try to run it using following steps in .\readme.txt and if it start skip next step;**
5. **Install and prepare latest Eclipse IDE for C/C++ developers (for Windows 64-bit):**
   a. Download Eclipse IDE from http://www.eclipse.org/downloads/;
   b. Rename C:\ChibiStudio\eclipse to C:\ChibiStudio\eclipse-32bit;
   c. Unzip Eclipse IDE zip file to C:\ChibiStudio\eclipse;
   d. Run it and install following plugins (work with All update sites):
      i. Add new update site embsysregview - http://embsysregview.sourceforge.net/update;
      ii. Eclipse XML Editors and Tools 3.6.3;
      iii. Embedded Systems Register View plugin 0.2.5;
      iv. EmbSysRegView Data plugin 0.2.5.r180;
      v. Target Management Terminal 3.7.0 (Deprecated) incl. RxTx Serial Connector plugin;
      vi. Install ChibiOS/RT plugins and features (use update site for latest versions if found):
         1. (org.chibios.*) from \eclipse-32bit\plugins to \eclipse\dropins\eclipse\plugins (4 plugins);
         2. (org.chibios.*) from \eclipse-32bit\features to \eclipse\dropins\eclipse\features (1 feature);
   e. Run ChibiStudio:
      i. Chose for Workspace one of the existing once (\workspace26 or \workspace30) or make new one;
      ii. In case of using existing workspaces all demo and test projects will be available in Project explorer ready for build and usage;
      iii. verify that all above plugins and features are available (Menu -> Help -> About Eclipse -> Installation Details);
6. **Add new ChibiOS version (from trunk for example):**
   a. Open http://sourceforge.net/p/chibios/svn/HEAD/tree/trunk/ and Download Snapshot;
   b. Unzip it in .\chibios302015032?? for example;
   c. Open https://github.com/ChibiOS/ChibiOS-Contrib and Download Zip;
   d. Unzip it in .\chibios302015032?\community\
   e. Unzip fatfs-0.10b-patched.7z and lwip-1.4.1_patched.7z files in .\chibios302015032?\community\ext\
7. **Add, build and deploy existing project from ChibiOS/RT Demos in new and empty Workspace**
   a. Change once following global preferences (from Menu -> Window -> Preferences):
      i. General -> Workspace -> Build automatically (uncheck);
      ii. C/C++ -> Code Analysis (uncheck all problems);
      iii. C/C++ -> Debug -> EmbSys Register View set Architecture: cortex-m4, Vendor: STMicro, Chip: stm32f40x, Board: none or one from the list (STM32F4Discovery only for the moment);
      iv. C/C++ -> New C/C++ Project Wizard -> Makefile Project -> Discovery Options -> Automate Discovery paths and symbols (uncheck);
   b. In Project Explorer -> Popup Menu -> Import… -> General -> Existing project into Workspace -> Next -> Select root directory -> Browse -> C:\ChibiStudio\chibios\demos\ARMCM4-STM32F407-LWIP-FATFS-USB -> OK where chibios can be one of the existing ChibiOS versions (stable 2.6.7, unstable 3.0 or 3.0 trunk) resided in C:\ChibiStudio folder;

2 pages (2015)

How to operate STM32Cube and ChibiOS based projects in unified environment composed by free and open source tools only

Project targets and references

The project target is to:

- Build combined application code base with ChibiOS and STM32Cube incorporating
  - USB CDC Host to Serial adapter like functionality
  - USB CDC Device to Serial adapter like functionality
  - USB vendor specific Host CDC like classes for CP201x and FT232R adapters
  - USB vendor specific Device CDC like emulation of CP201x and FT232R adapters
  - Performance testing and application debugging staff
- Create development environment based on Eclipse CDT, GNU Arm and Makefiles
- Test a way and describe the path
  - for doing above from the scratch
  - using only free and open source tools and code base

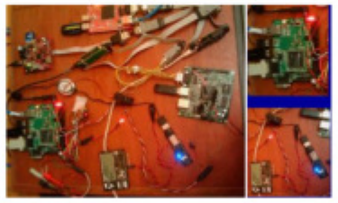
Startup references:

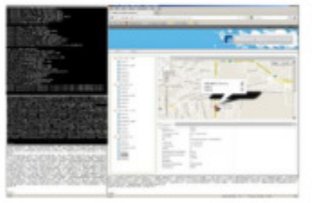
- Eclipse IDE and community projects http://www.eclipse.org/
- GNU tool chain for ARM Cortex-M/R processors https://launchpad.net/gcc-arm-embedded
- GnuWin32 binutils and management tool for Windows http://gnuwin32.sourceforge.net/
- ChibiStudio, ChibiOS, its demos and community staff from: http://chibios.org/
- STM32Cube application completely generated by STM32CubeMX for STM32-F407 board from
  - http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/LN1897/PF259243
  - http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259242?s=micoxplorer
- Experience to create from scratch Eclipse based environment for ChibiOS
- Makefile based sample project for STM32-F407 board from http://wunderkis.de/stm32cube/index.html
- Demo projects including all described implementation
  - at the moment: ab-tests.rar, atest-L.rar, atest-2.rar and ab-tests-howto.pdf (this file)
  - can be downloaded from http://www.radevs.net/~cbradev/projects/ab-tests/

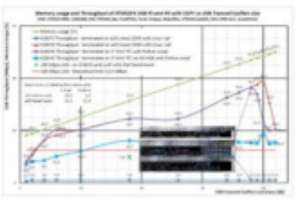
Project stages done:

- Created development environment - single installation to work with:
  - Eclipse Mars CDT 32-bit version on Windows 7 with Java JDK 1.8 both 64-bit versions
    - installed Eclipse IDE for C/C++ developers with Mylyn integration
    - added plugins: STM32CubeMX, GNU Arm C/C++, ChibiOS, EmbSysRegView, TM Terminal, RxTx, SVN etc.
    - tested and tuned global and project settings to work with STM32CubeMX, GNU Arm C/C++ and ChibiOS
  - GNU tool chain from ARM Cortex-M/R MCUs from Launchpad for Windows version 150626
  - GNU tools from ChibiStudio, binutils GnuWin32 complete set, QEMU, OpenOCD ver. 0.9, UVCview etc.
- Created ChibiOS project based on USB CDC Demo for STM32-H405
  - added board support for STM32-H405 to ChibiOS STM32 based boards
  - ChibiOS USB CDC Demo for STM32-H407 ported to STM32-H405
- Created STM32CubeMX project based on STM32-F407 IOC file from wunderkis.de
  - added all peripherals support incl. FatFS, FreeRTOS and LwIP middleware
  - integrated Makefile file set to generated project code base from wunderkis.de
  - tuned to fit complete STM32-F407 board specification
- STM32Cube application completely generated by STM32CubeMX for STM32-F407 board
  - created and tested basic application without any application implementation
  - added standard IO redirection to USART for debugging
  - added floating point support using STM32 FPU
  - tested multi device support (CDC and MSC Host classes)
  - tested a way and added more FreeRTOS tasks
  - extended STM32Cube based project with USB Host CDC to Serial functionality
  - added a simple bidirectional data transfer between USB Host CDC and USART
  - added support and measured bandwidth with ChibiOS application on STM32-H405 board
  - extend STM32Cube based project with USB Device CDC functionality
  - added transmitter simulating massive data stream from USB HS CDC Device to USB FS CDC Host
  - added code to calculate and print test statistics
- Make performance tests for throughput, packet utilization and data loss
  - throughput is measured at different buffer sizes, without and with UART and at different UART bitrates
  - packet utilization is measured as 100% in case of sending buffer with size proportional to packet's one
  - there is no data loss measured as a difference between data sent by USBH and retransmitted by USBH sides

17 pages (2015)

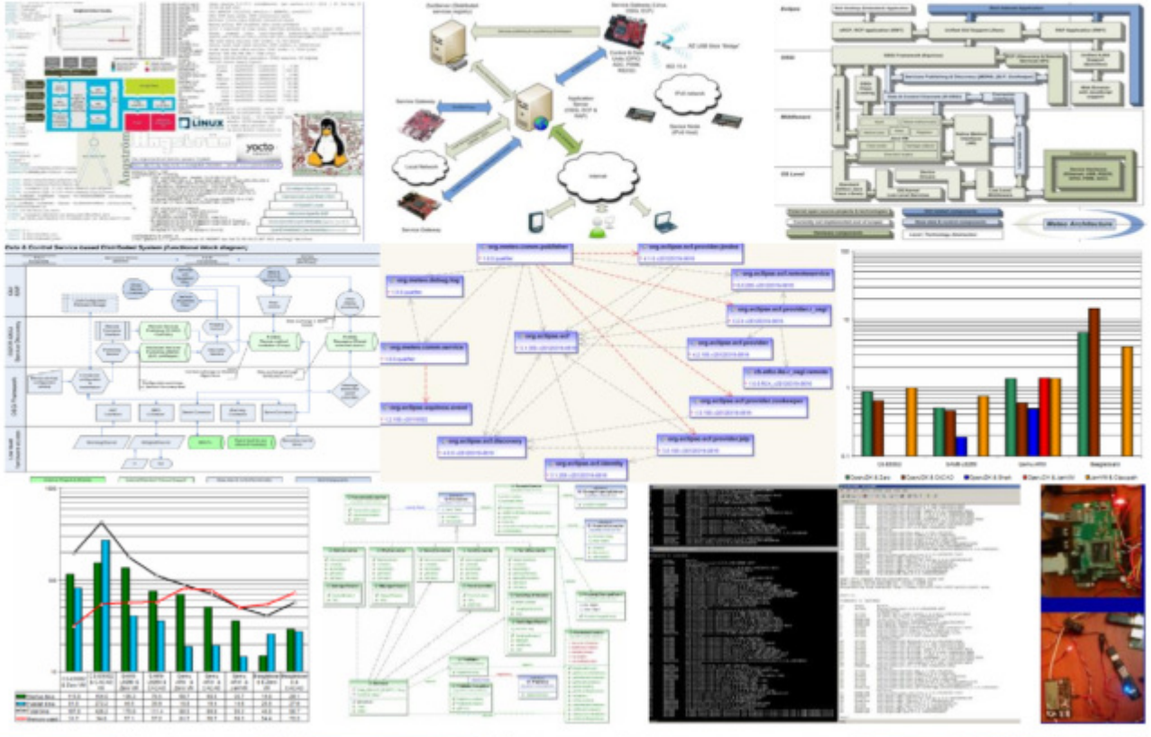




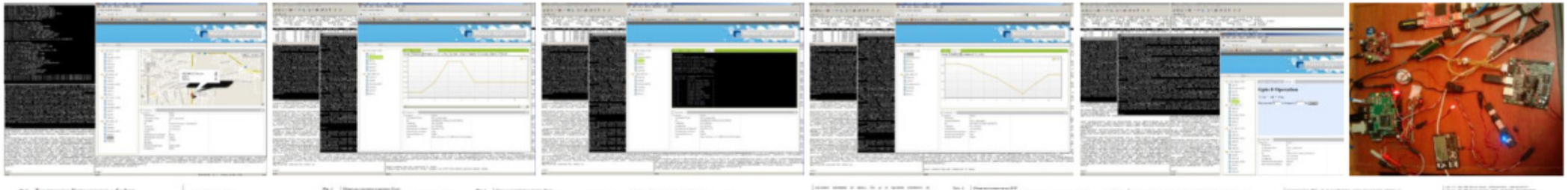


When working with a student on his thesis number of How-To documents was written for different aspects of the corresponding projects like the process of building and setting up Ångström distribution from scratch, installing ChibiStudio and Eclipse as development IDE on different OS platforms, usage of ChibiOS for real time embedded projects. One of the very useful documents was description of how to operate STM32Cube and ChibiOS based projects in unified environment composed by free and open source tools only.

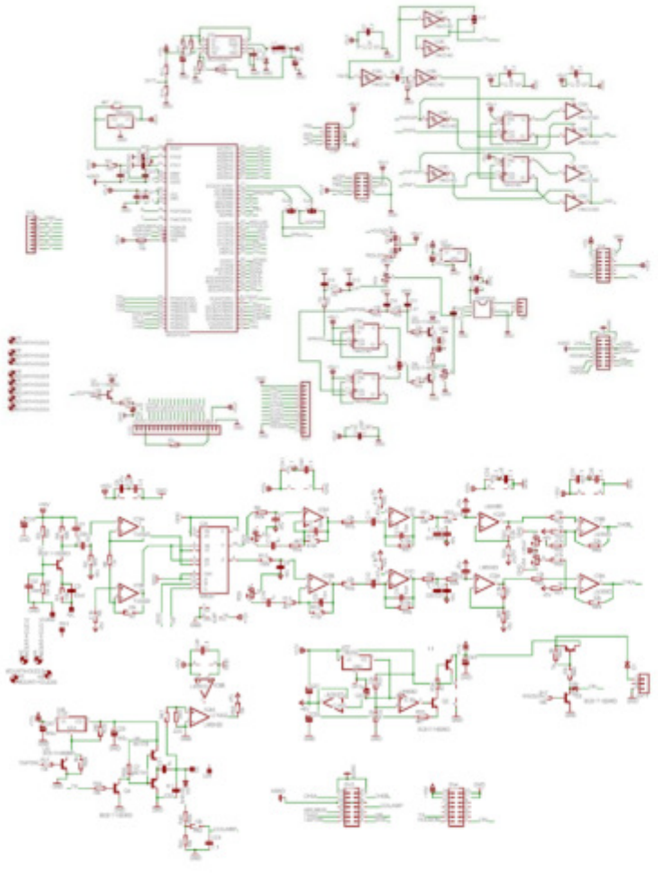
Student's thesis: master degree (2010 – 2012)
Subject: OSGi based distributed sensor system

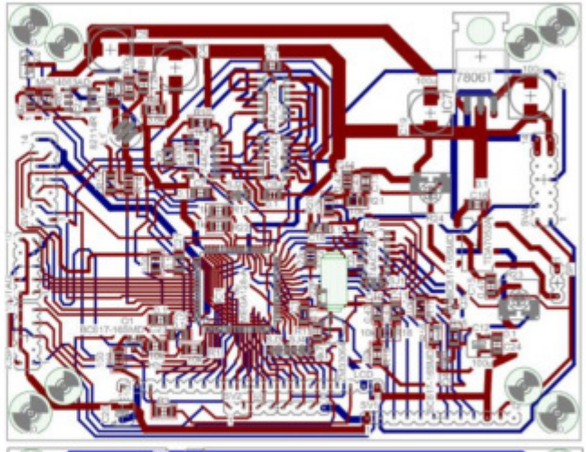


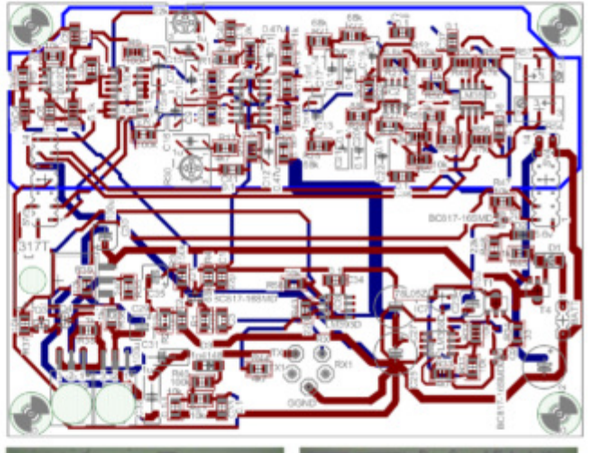
The spots of openness influenced the work



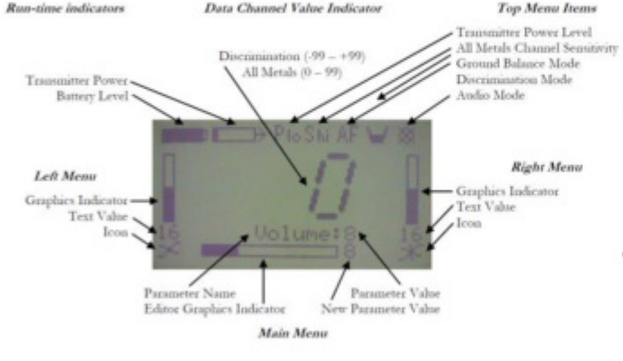
GMD Ltd.: VLF Metal detector based on Mega128/uSmartOS with gLCD/GUI (2010 – 2011)
Development, programing, testing, introduction: Embedded hardware and software implementation



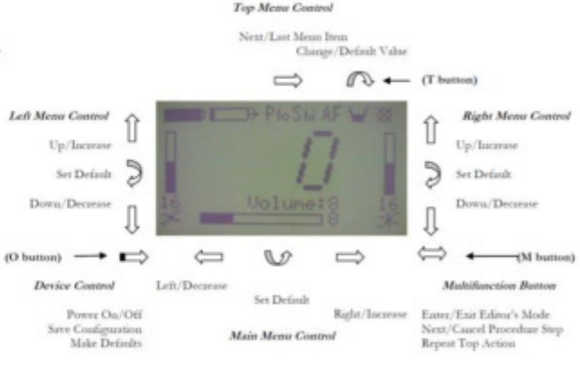




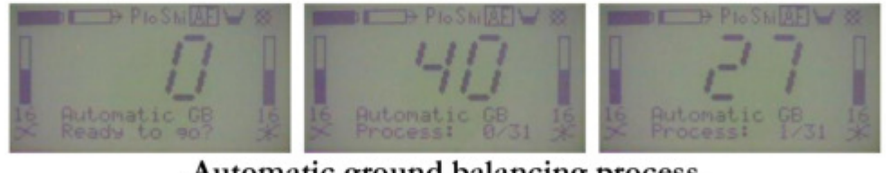
GUI (Graphics User Interface) description



MMI (Machine/Man Interface) description

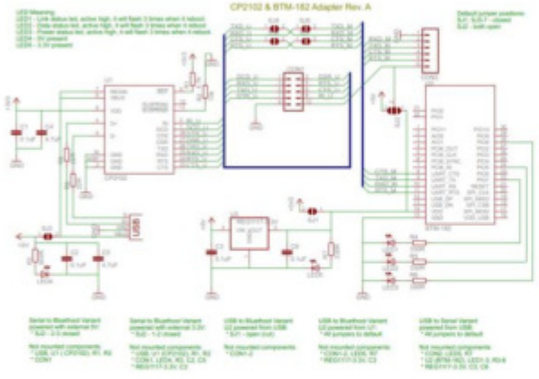


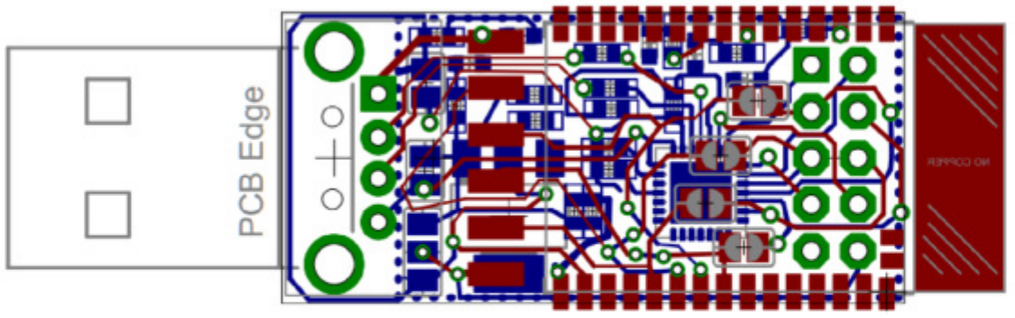
Firmware start up procedures



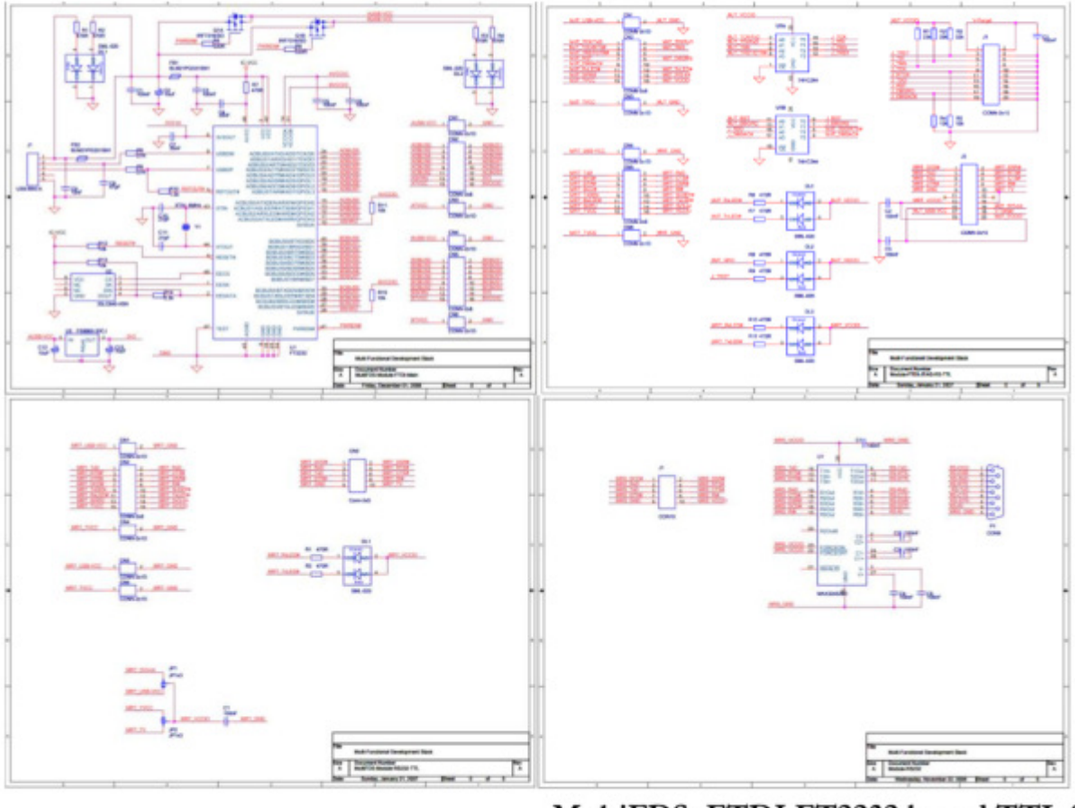
Automatic ground balancing process

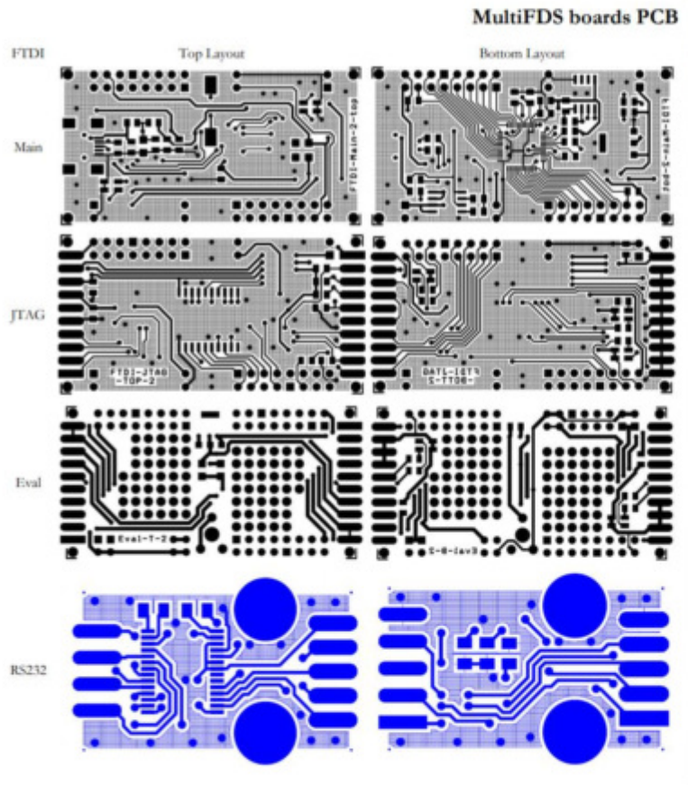
Helpful hardware projects (2007 – 2015)
Development, production, testing: Hardware implementation





Multifunctional wired/wireless serial cable – USB/Bluetooth ⇔ Serial (TTL/RS232) adapter based on BTM-182 and CP2102 (2015)



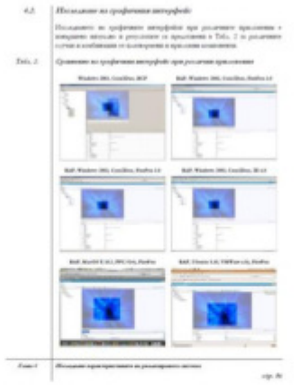


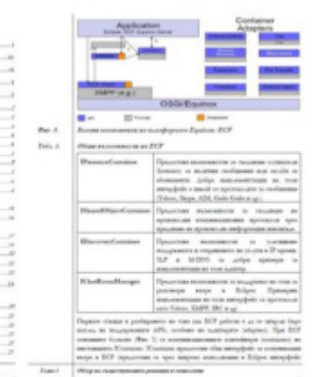
MultiFDS: FTDI FT2232 based TTL Serial/RS232/JTAG adapter (2007)

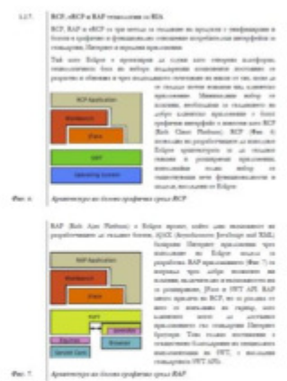
Student's thesis: bachelor degree (2007 – 2009)

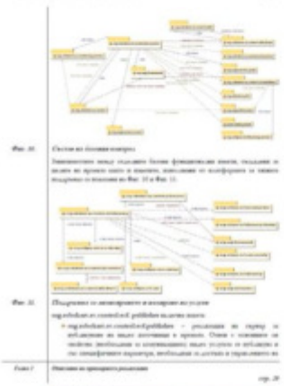
Subject: Management of video streaming in a distributed system for video surveillance

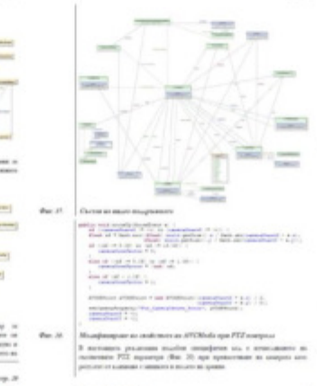
Participation in Eclipse DemoCamp, 2008, Sofia

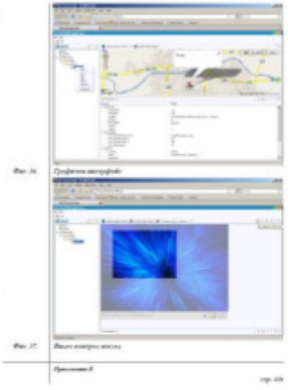


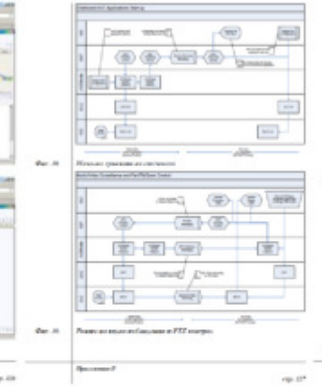




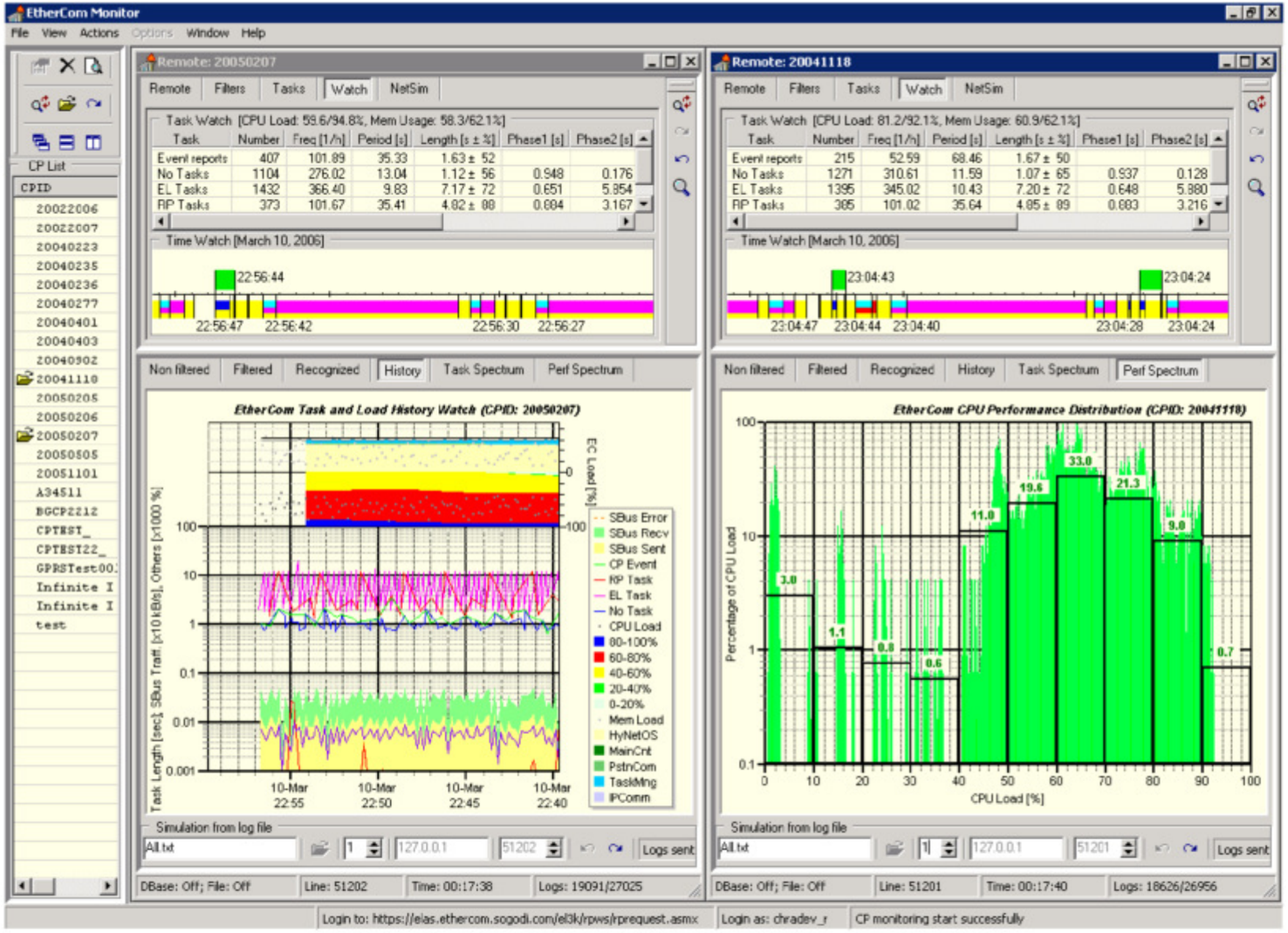




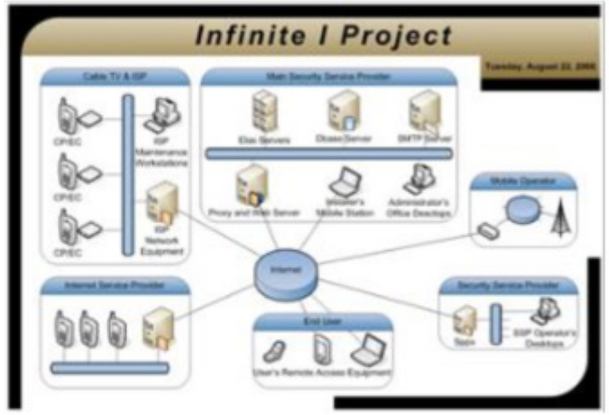


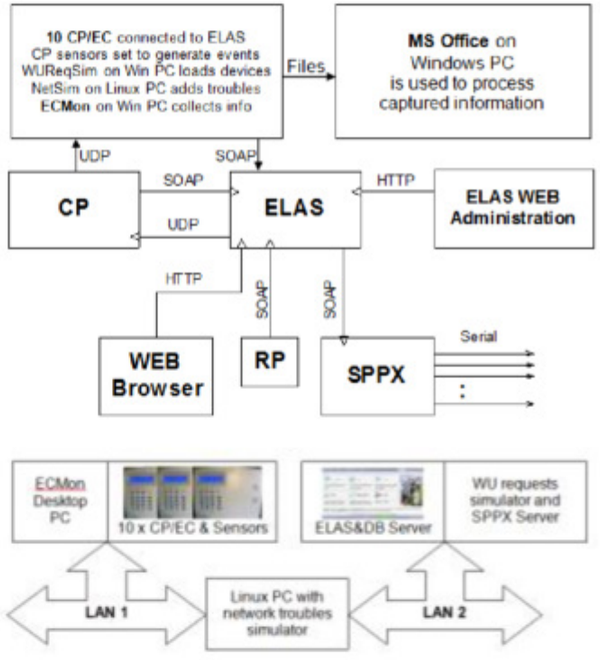


Electronics Line 3000: Internet enabled security system (2003 – 2008)
Introduced: SOA and Long term high load functional testing methodology



Snapshot of real test case





Student's thesis: bachelor degree (2002 – 2003)
Subject: Internet accessible embedded system for distance measuring

Master Project Presentation

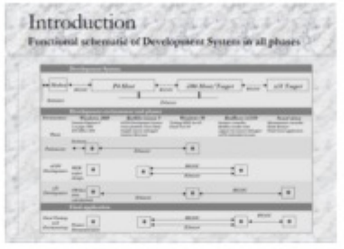
Internet accessible embedded system for distance measuring

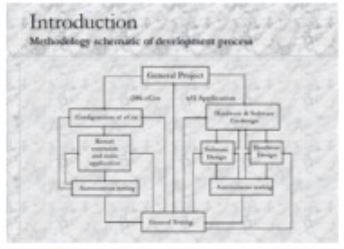
Author: Dilyana Panayotova
Technical University - Sofia, branch Plovdiv,
Department Electronics and Automation

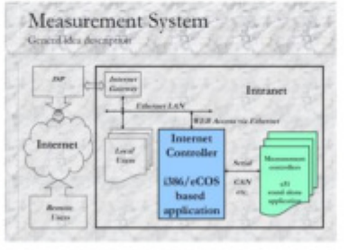
Supervisor:
M.Sc. Dr. Valentin Mollov
Head of CE Department

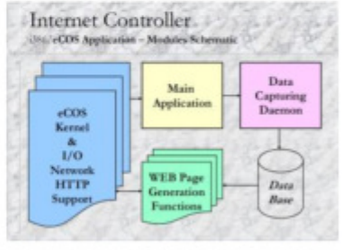
Consultant:
Christo Radev
CS/IT Expert

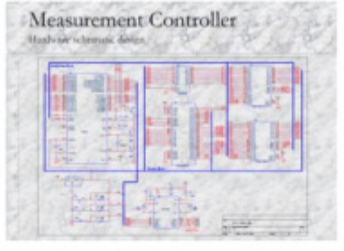
Referee:

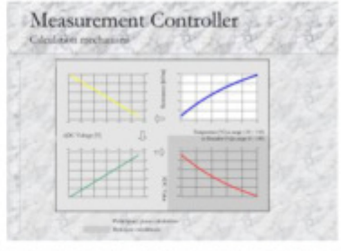


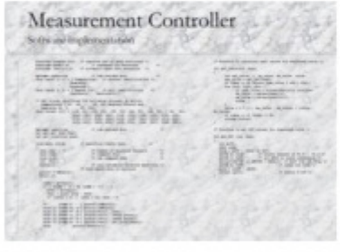


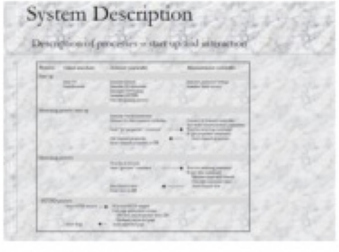


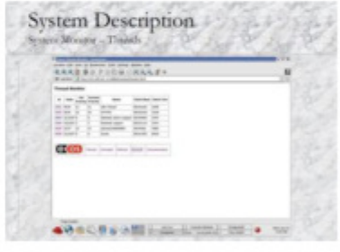


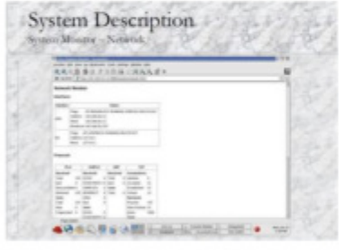


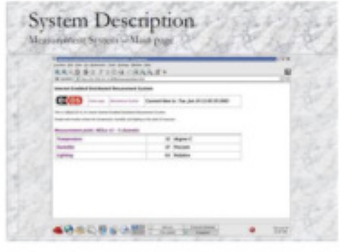


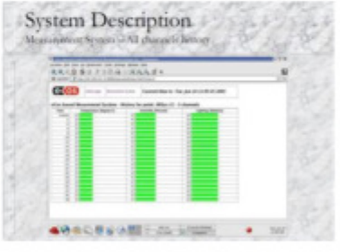




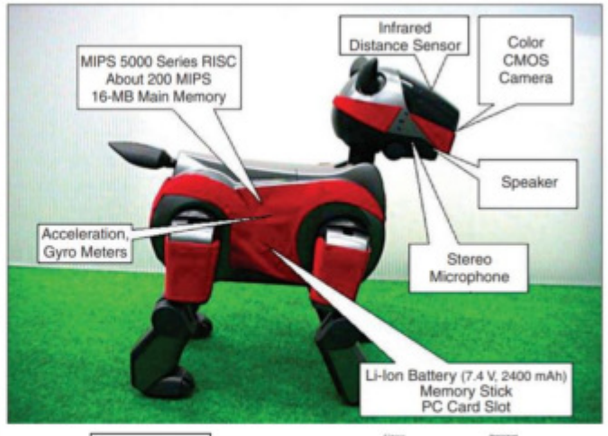


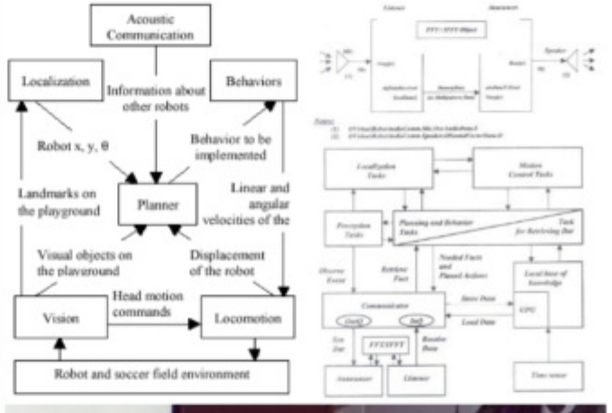






TU Plovdiv, BU Istanbul, Sony: Autonomous soccer playing AIBO robots (2000 – 2001)
RoboCup 2001 (Sony Four-Legged Robots League) participation: Balkan (Cerberus) international team

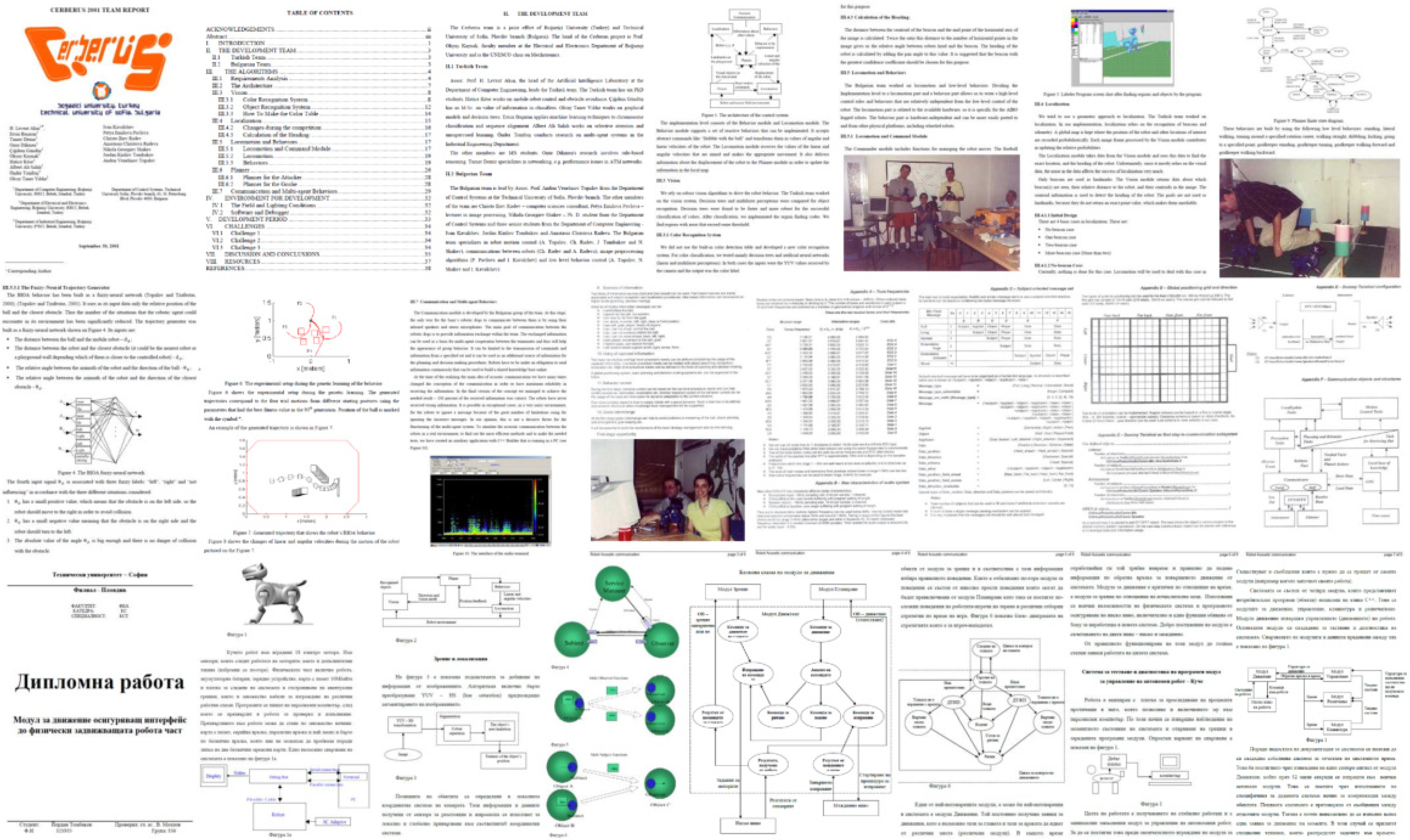




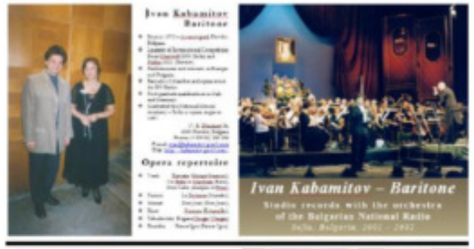
Real football match between AIBO robots (RoboCup 2001 Final)

TU Plovdiv, BU Istanbul, Sony: Autonomous soccer playing AIBO robots (2000 – 2001)

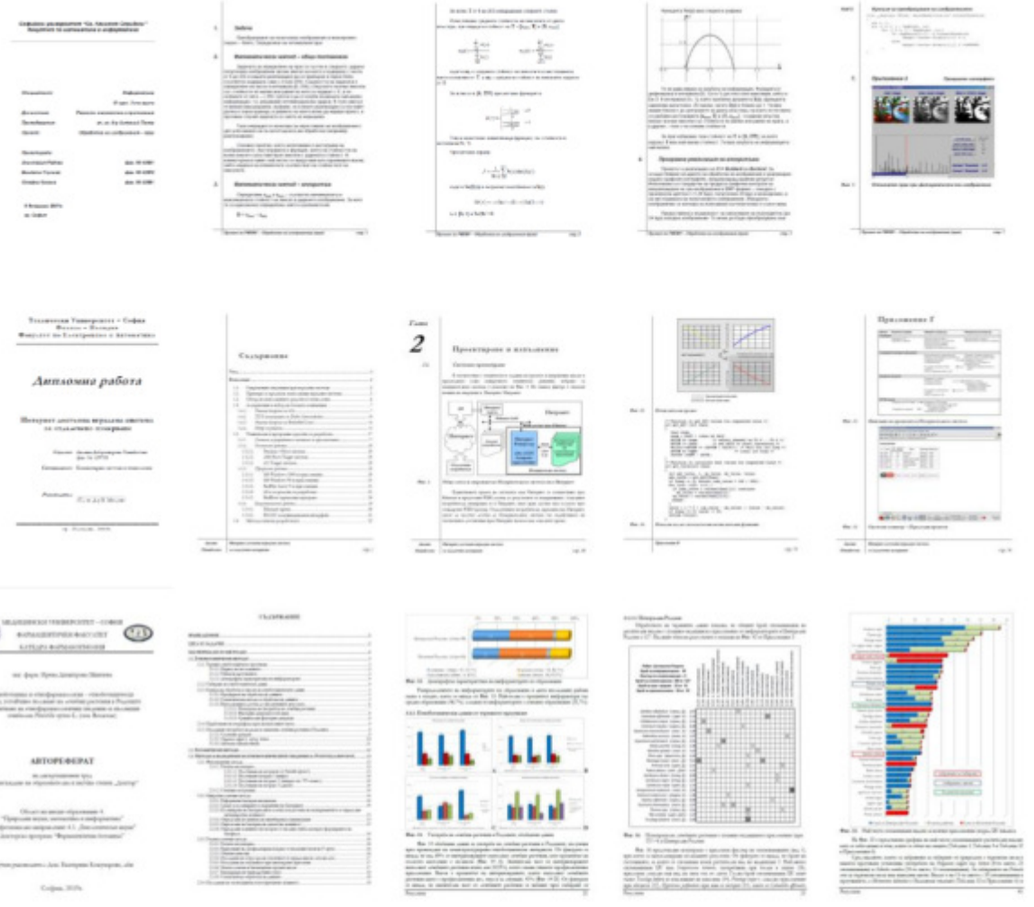
Bulgarian team share: Acoustic communications, Behaviour, Locomotion, Multi-agent behaviour and System control



Multimedia, paper, web and presentations design projects (1998 – 2019)



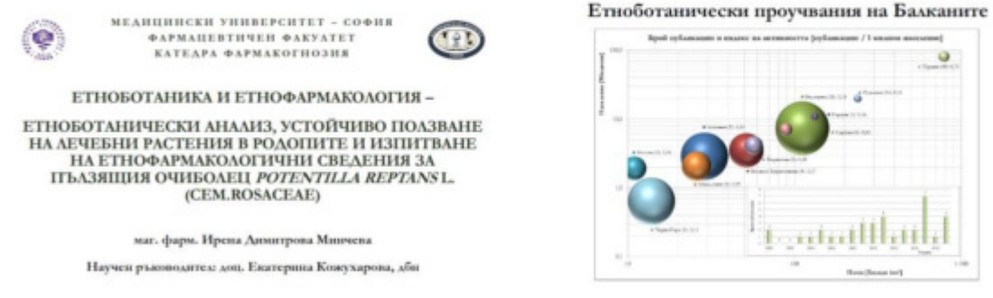
Personal and professional presentations: multimedia and audio CDs



Project reports, graduation works, doctoral dissertations etc.

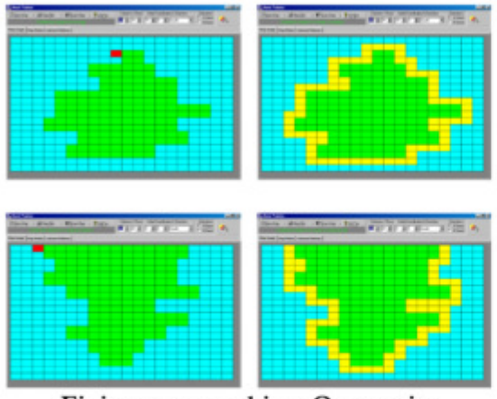
Cerberus team printed leaflet

Professional web sites

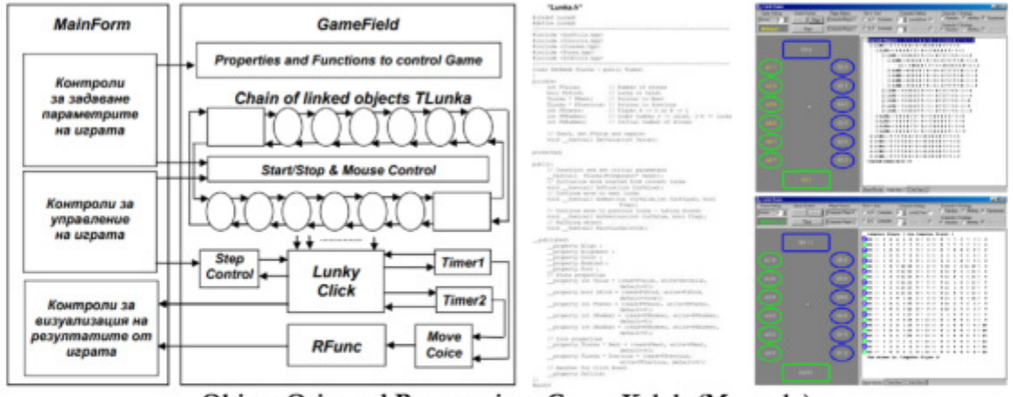


Professional presentation of doctoral dissertations

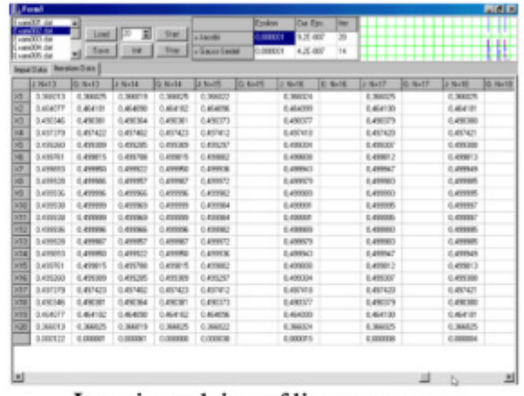
Student's projects and graduation works guidance (1998 – 2003)



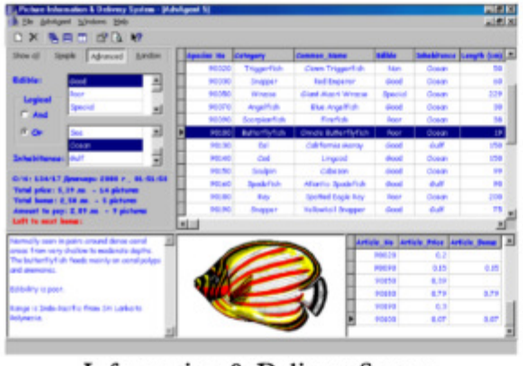
Finite-state machine: Ocuntorite



Object Oriented Programing: Game Kalah (Mancala)



Iterative solving of linear systems



Information & Delivery System

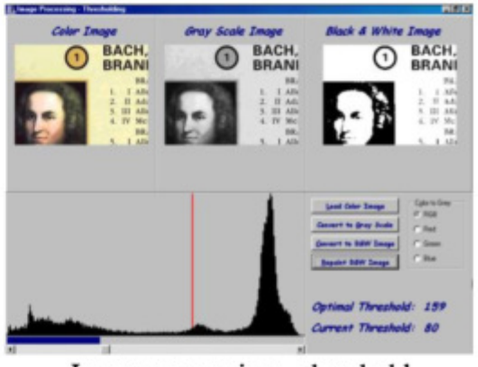
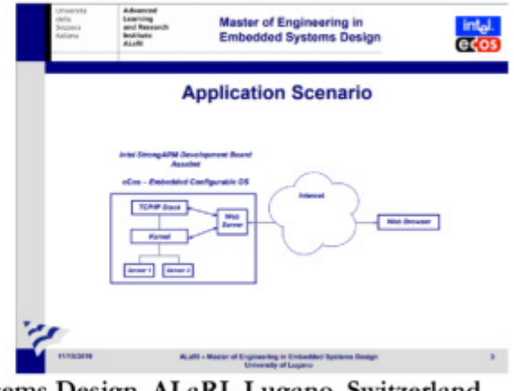
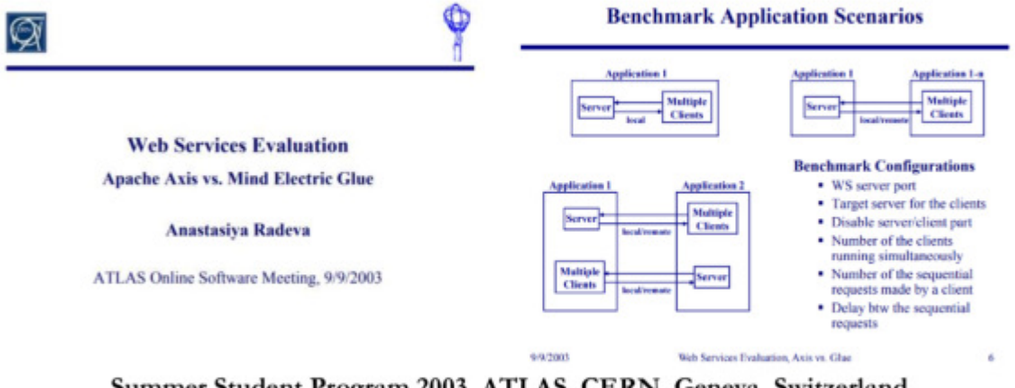


Image processing – threshold

Laser diode projector (micro controller programming, project builder and programmer)

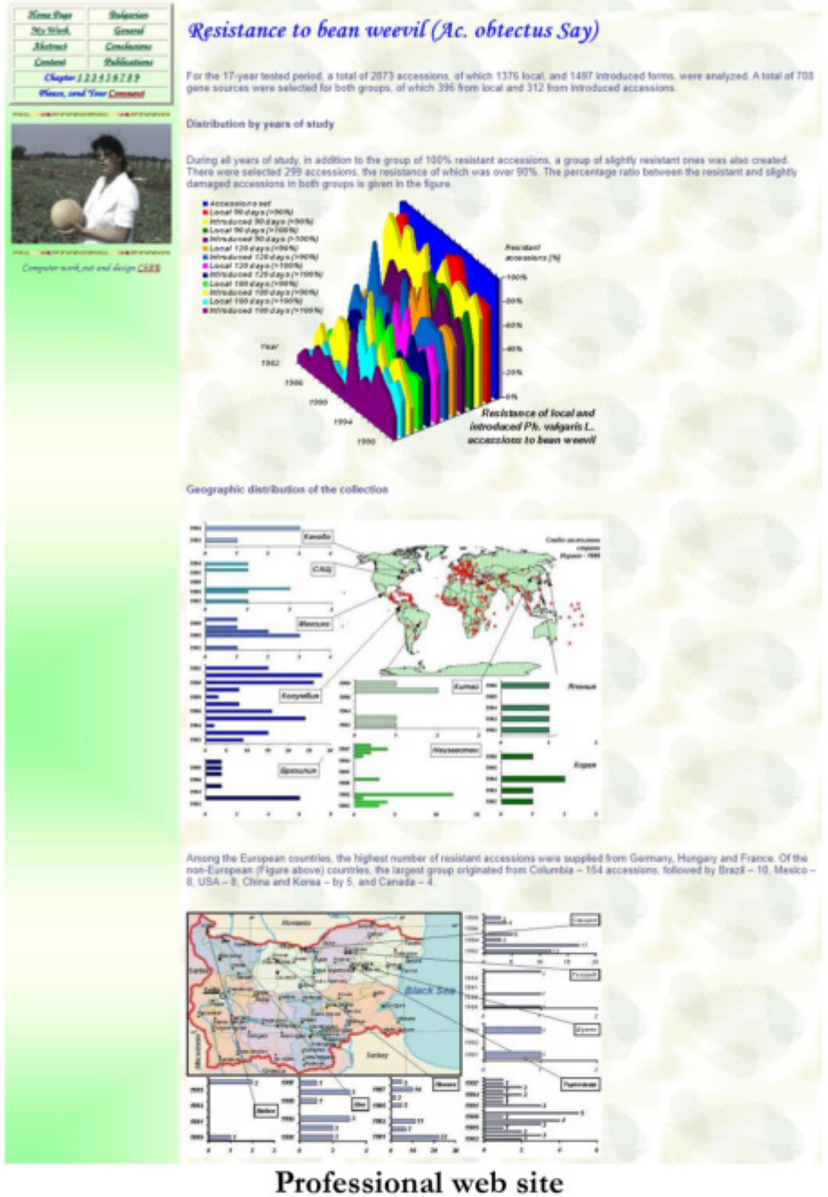


Master of Engineering in Embedded Systems Design, ALaRI, Lugano, Switzerland

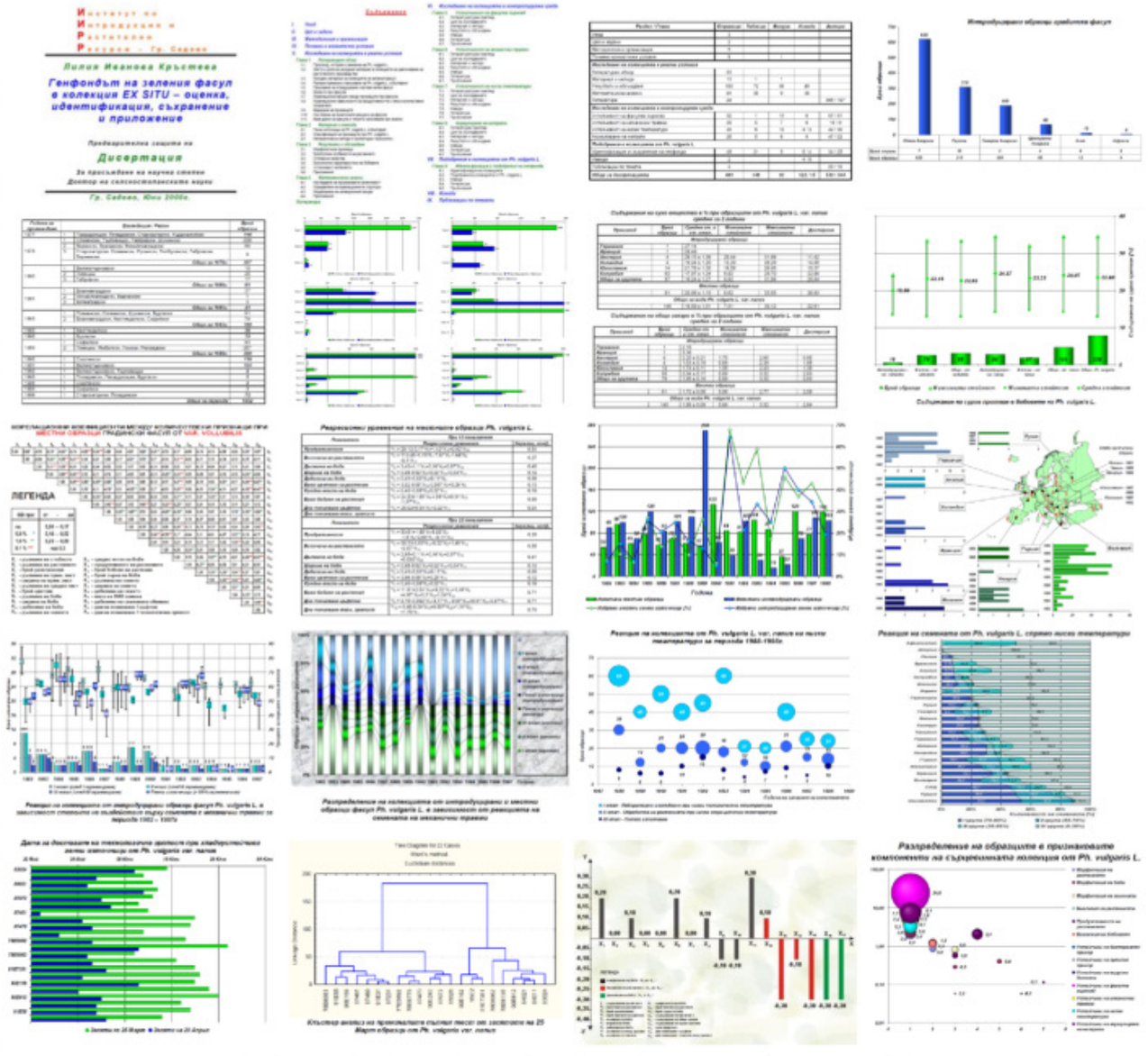


Summer Student Program 2003, ATLAS, CERN, Geneva, Switzerland

Doctoral dissertation: mathematical support, paper and presentation design and work out (1998 – 2000)
Subject: Green Bean Germplasm in an EX SITU Collection – Evaluation, Identification, Conservation and Use

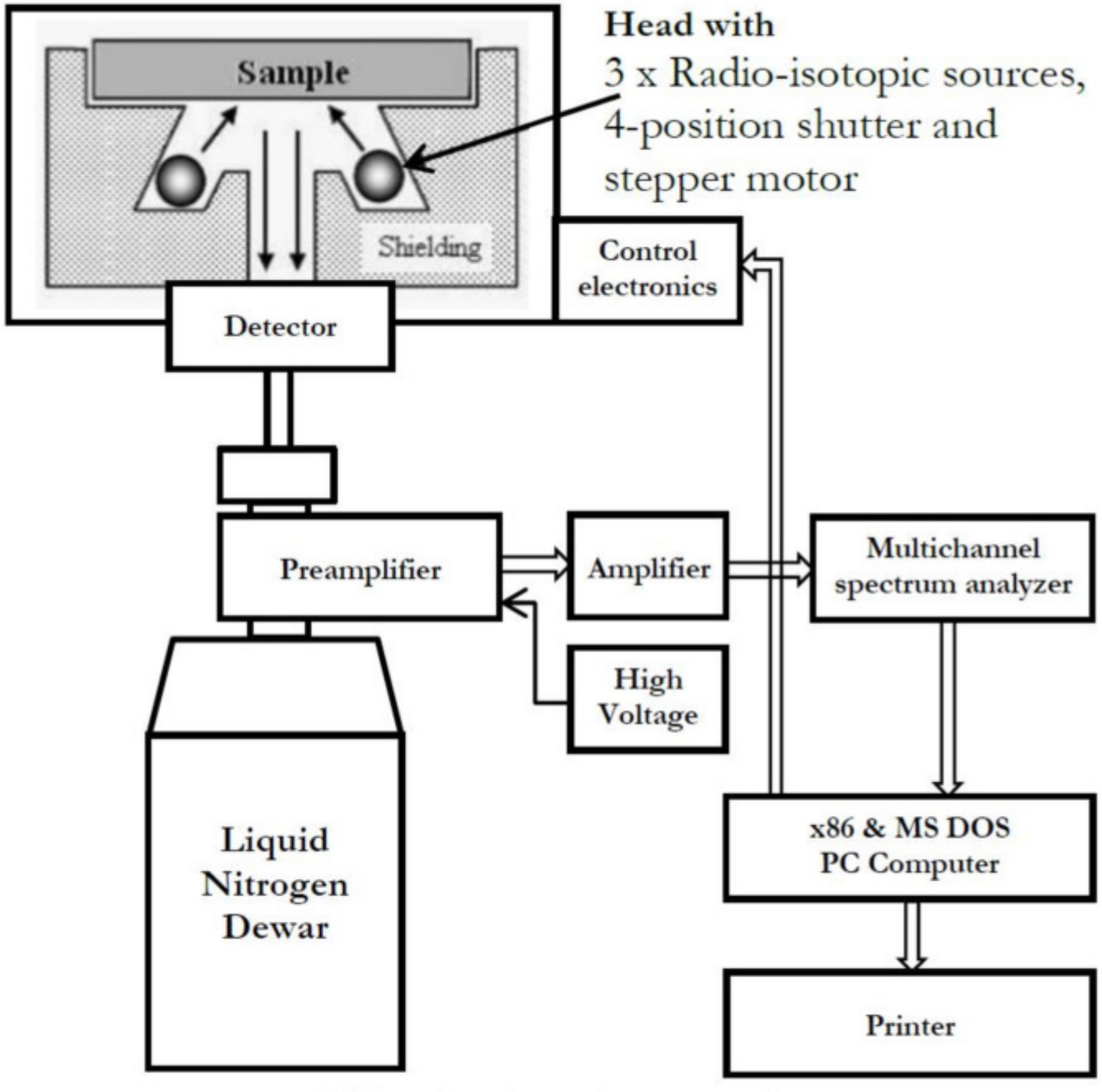


Professional web site



Mathematical results, printed and presentation materials

G.L.T. Ltd., Plovdiv: System for Roentgen-Fluorescence Analysis of heavy metals concentration (1996 – 1997)
Development, programing, testing, introduction: Embedded and desktop software implementation



RFA work station and computer suit

Program for optimization of RFA equations

Non-ferrous metal mill JSC - Plovdiv

X-ray Fluorescence Laboratory
24/2/97 - 12:23:37

Lead production
Material: Lead concentrate
System recalibration started on: 7/2/97

Coefficients of the equation for: Zn

	Coefficient * Parameter
Zn % =	-2.227568
	+0.284827 * Zn
	-0.000791 * Zn*Bg

Correlation coefficient R = 0.9721
Dispersion S = 0.3248

Changed intensities of control standards

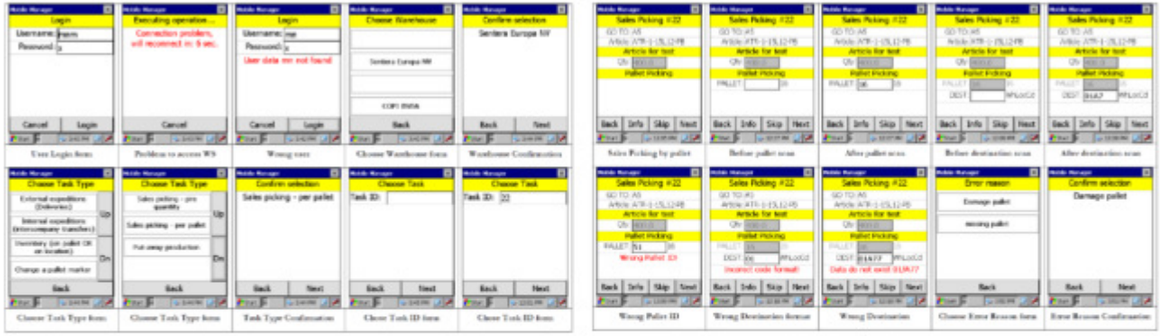
№	Sample	Chem. analysis	R F A	Diversion
1	413	6.930	6.737	-0.193
2	474	7.980	8.191	+0.211
3	477	8.900	8.682	-0.218
4	487	7.450	7.531	+0.081
5	614	9.330	9.037	-0.293
6	439	6.250	6.769	+0.519
7	533	6.330	5.972	-0.358
8	421	6.500	6.594	+0.094
9	467	8.420	8.884	+0.464
10	472	8.290	8.175	-0.115
11	488	8.690	8.402	-0.288
12	519	7.680	7.776	+0.096

Operator: _____

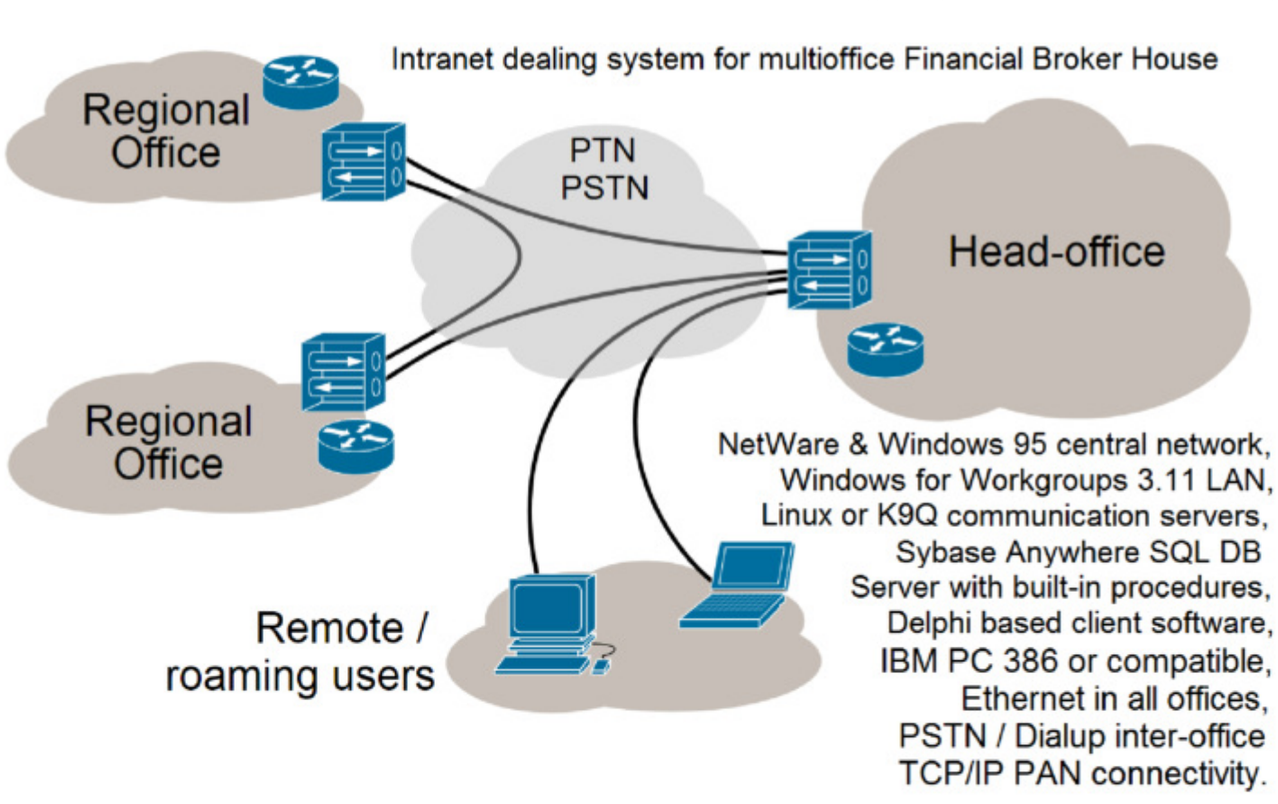
Protocol for system recalibration

Nova Ltd., Plovdiv: Office Systems (1994 – 2003)

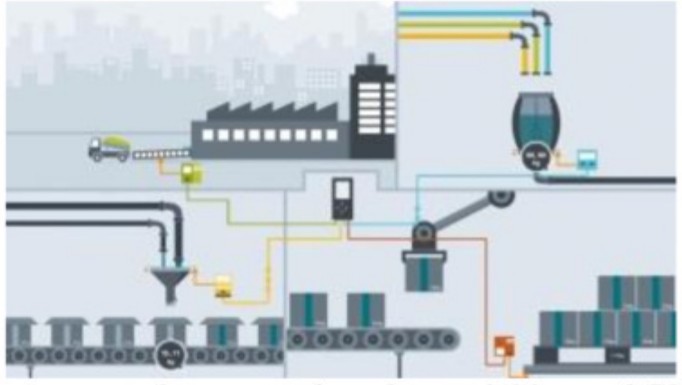
Development, programing, testing, introduction: Desktop and embedded software systems



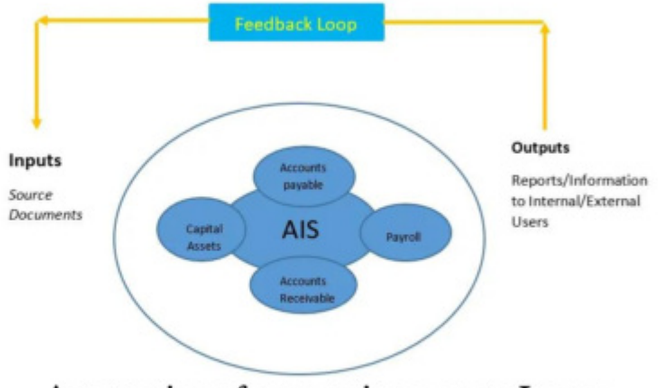
ERP system for low temperature warehouses based on barcodes, Progress Application Server (Windows 2003 / 7) and mobile application for Psion Teklogix 7530 G2 device (Windows CE 5.0 & .NET CF 2.0)



Multi-office Intranet system for selling in real-time



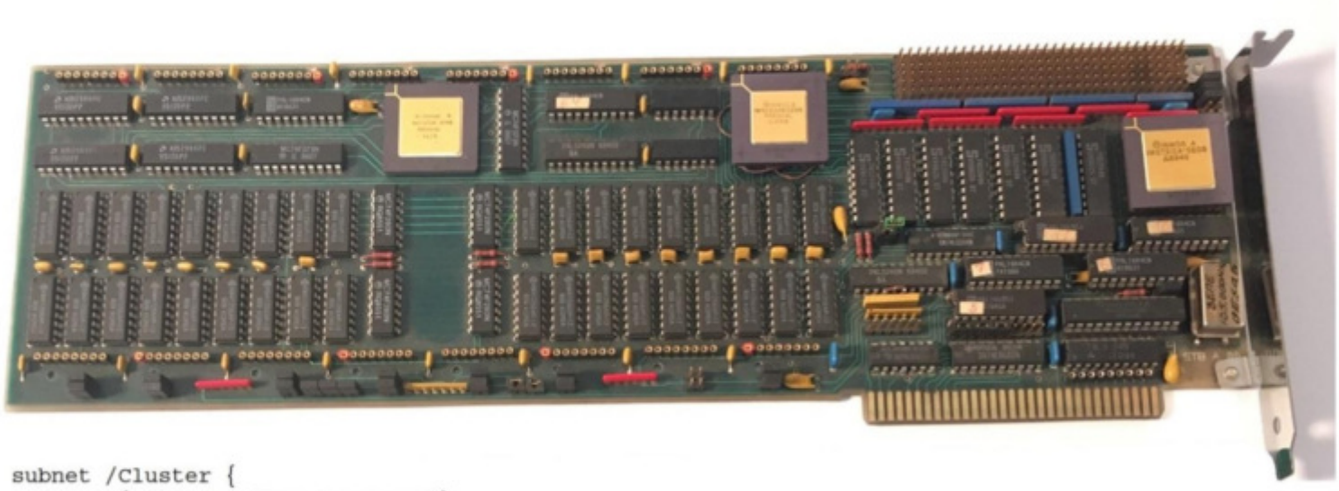
Plant automation system based on weighing and QNX



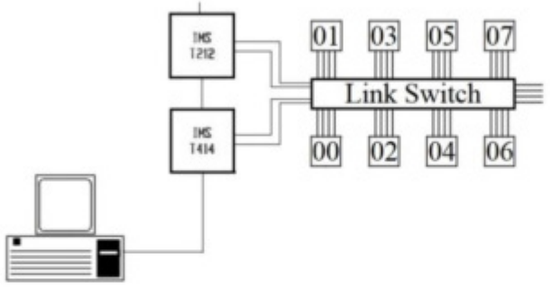
Automation of accounting system Logos

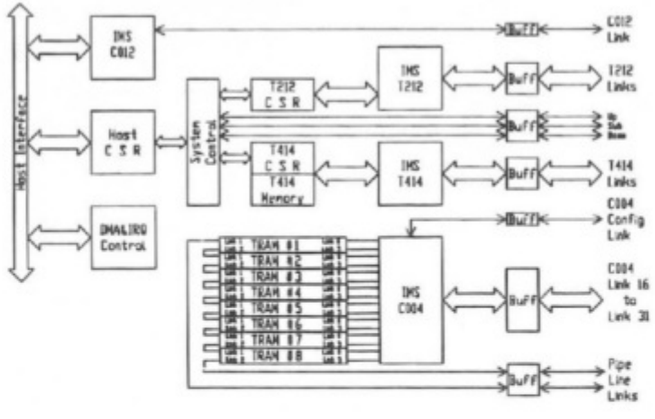
Nova Ltd., Plovdiv: Powerful computing system based on Inmos Transputers (1989 – 1994)

Development, production, testing, programing and application: PC board, Windows server, Lynx-T language, IAEA calculations



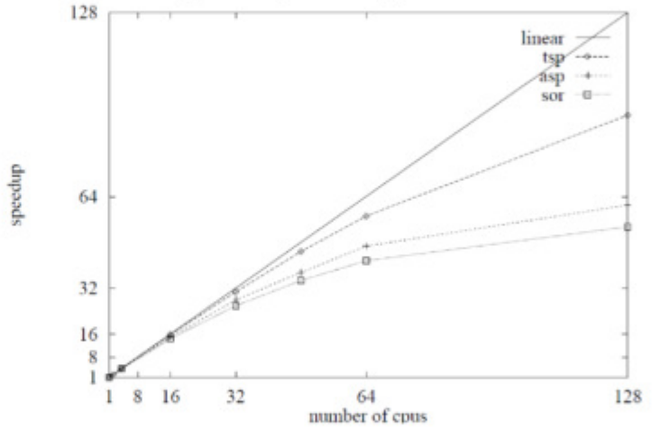
```
subnet /Cluster {
  Reset { driver; ~00; tram_ra.d}

  processor 00 { ~IO,    , ~01, ~02; }
  processor 01 { ~00,    ,    , ~03; run -e /helios/lib/fs fs scsi; }
  processor 02 {    , ~00, ~03, ~04; run /helios/lib/lock; }
  processor 03 { ~02, ~01,    , ~05; }
  processor 04 {    , ~02, ~05, ~06; }
  processor 05 { ~04, ~03,    , ~07; }
  processor 06 {    , ~04, ~07,    ; }
  processor 07 { ~06, ~05,    ,    ; }
  processor IO { ~00; IO }
}
```



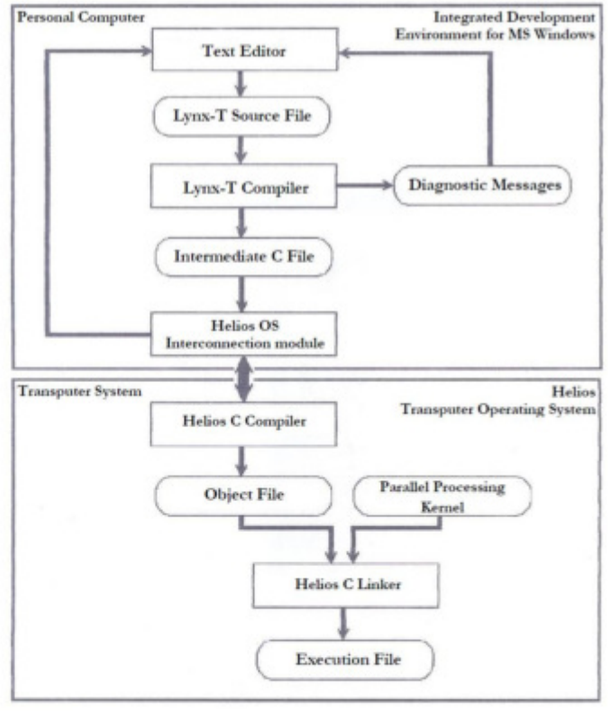


PC board hosting up to 10 transputers and programmable link switch

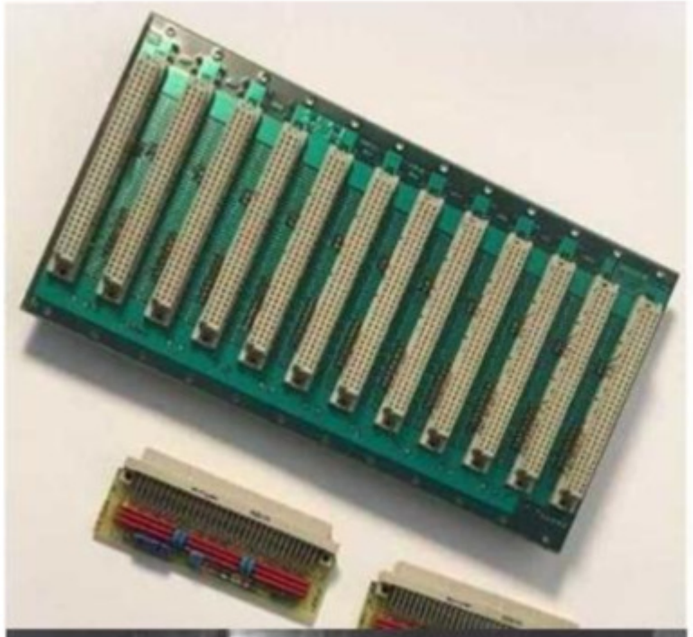
Transputer systems performance

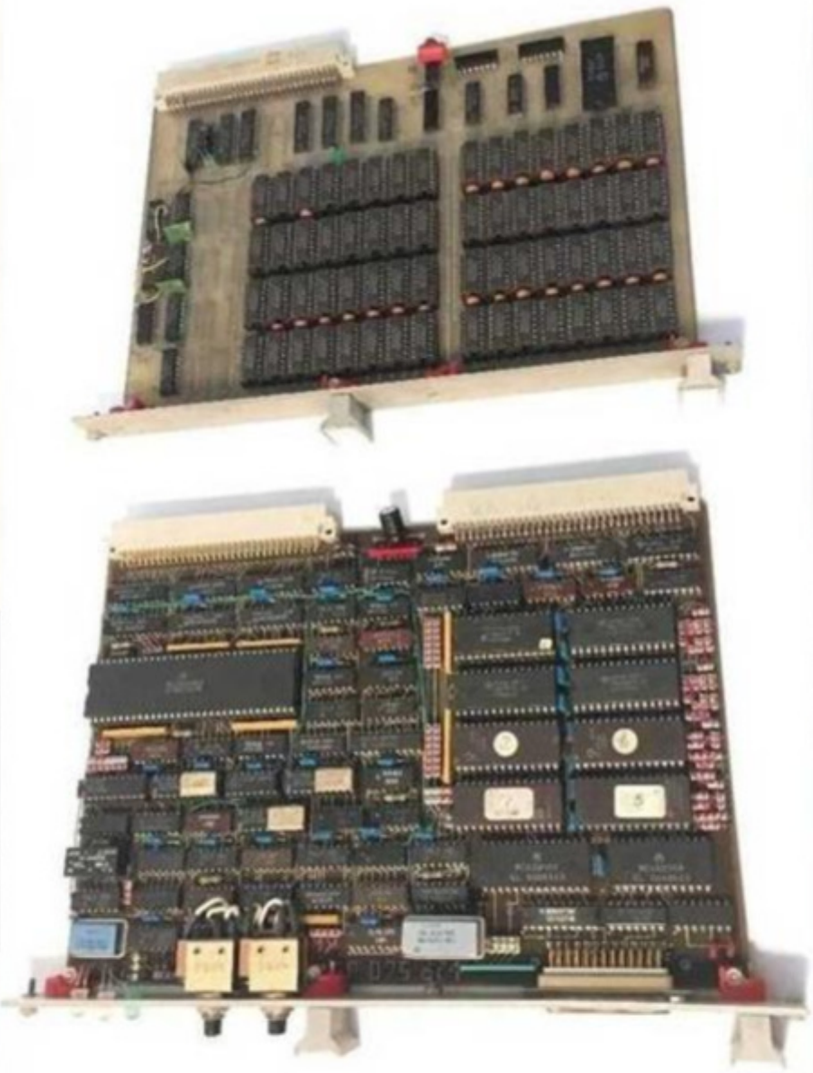


Language, tools and system for massive parallel computing

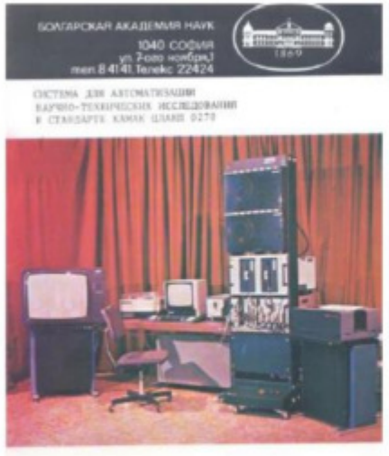


Bulgarian Academy of Science, Central Laboratory on Automation and Scientific Instrumentation, Plovdiv (1983 – 1988)
Development, production, testing and programing: VME-bus Instrumentation System based on MC68000 (Interlab)





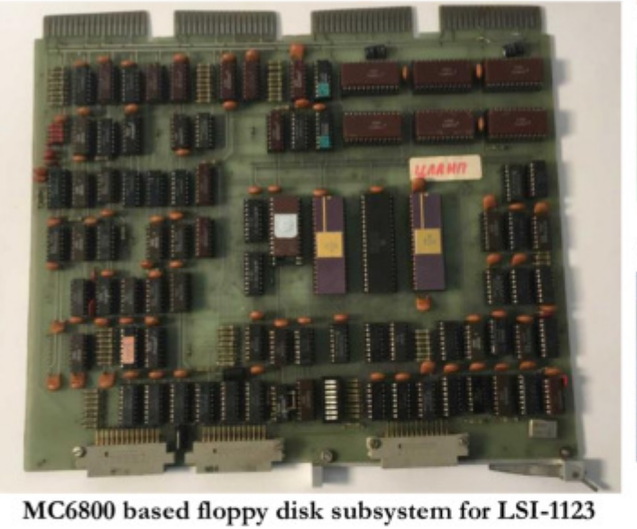
Bulgarian Academy of Science, Central Laboratory on Automation and Scientific Instrumentation, Plovdiv (1983 – 1988)
Development, production, testing and programing: CAMAC Instrumentation Systems



Basic system in CAMAC standard based on MC6800 – CLASI 270

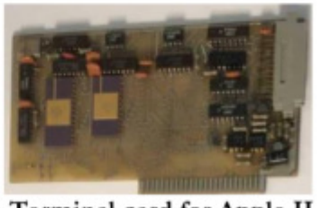
Basic system in CAMAC standard based on LSI-1123 – CLASI 290

IEEE 488 for MC6800 based CAMAC module



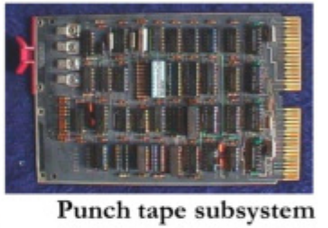
MC6800 based floppy disk subsystem for LSI-1123

MC6800/MC6840 based music CAMAC module



Terminal card for Apple II

Bulgarian singing Apple II



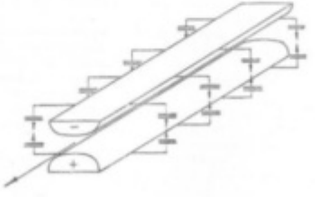
Punch tape subsystem for LSI-1123 I/O

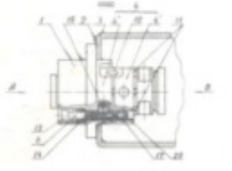
Bulgarian Academy of Science, Laboratory of Applied Physics, Quantum Electronics Department, Plovdiv (1979 – 1983)
Design, implementation, investigation, application: Powerful pulse-periodic TEA CO_2 Laser

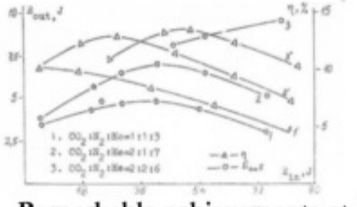
Industrial TEA CO_2 Laser 1300-M

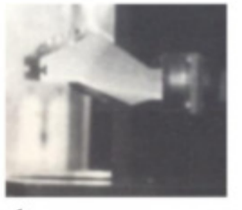
TECHNICAL PARAMETERS:

Wavelength	- 10.6 microns,
Pulse power (maximum)	- 100 MW,
Impulse energy (maximum)	- 6 J,
Pulse repetition rate	- 0.1 - 50 Hz,
Average power (maximum)	- 50 W,
Pulse duration	- 0.06 - 2 μs,
Gas consumption (CO_2: N_2: He = 1: 1: 8)	- 1 l / min,
Water consumption	- 10 l / min,
Power supply 3 x 380 V / 50 Hz	- 2.5 kW,
Laser head size	- 1100/880/510 mm,
Power supply unit size	- 1600/900/700 mm,









Remarkable achievements at development process

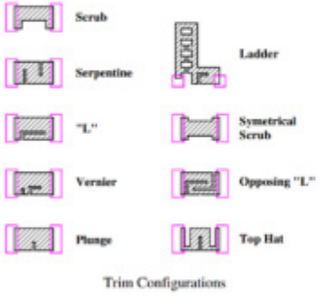
Industrial TEA CO_2 Laser
XI National Exhibition

Laser marking system
XII National Exhibition

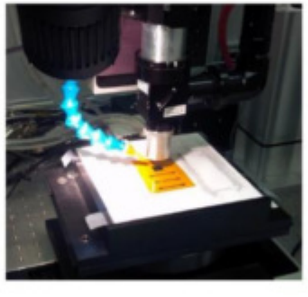
Golden medals from Technical and Scientific Creation of Youth

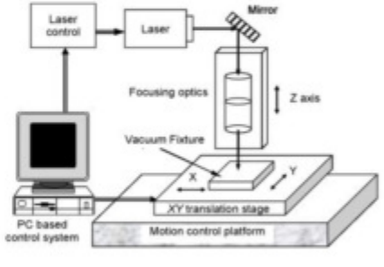
Bulgarian Academy of Science, Laboratory of Applied Physics, Quantum Electronics Department, Plovdiv (1979 – 1983)
Design, implementation, programing and application: MC6800 based control system for laser applications

Prototypes of MC6800 based control system and powerful TEA CO_2 Laser



Thin films trimming

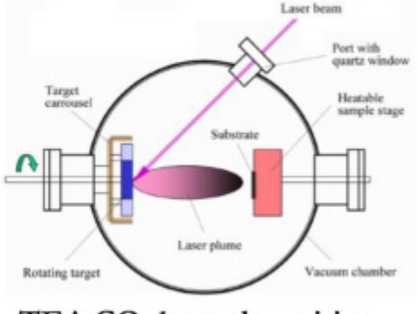




CO_2/N_2 laser tuning, scribing, engraving etc.

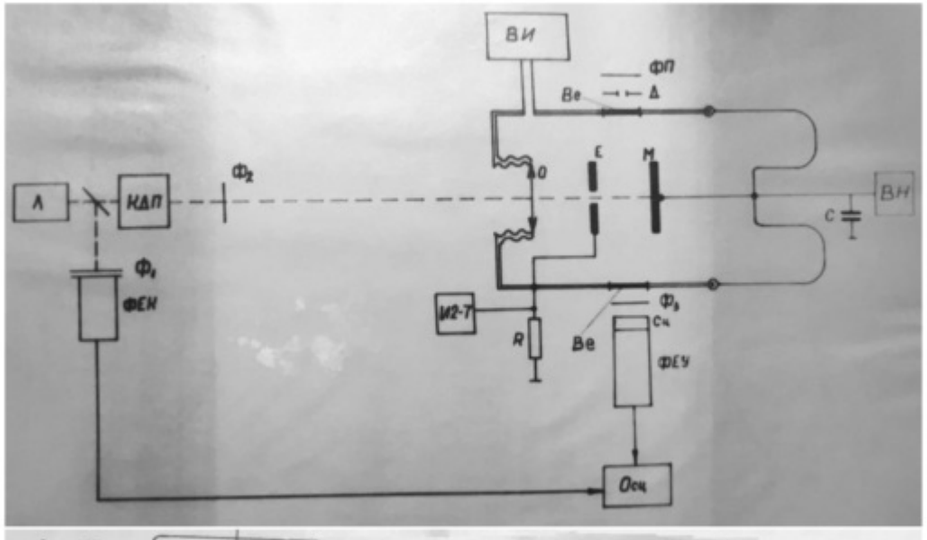
Glass marking

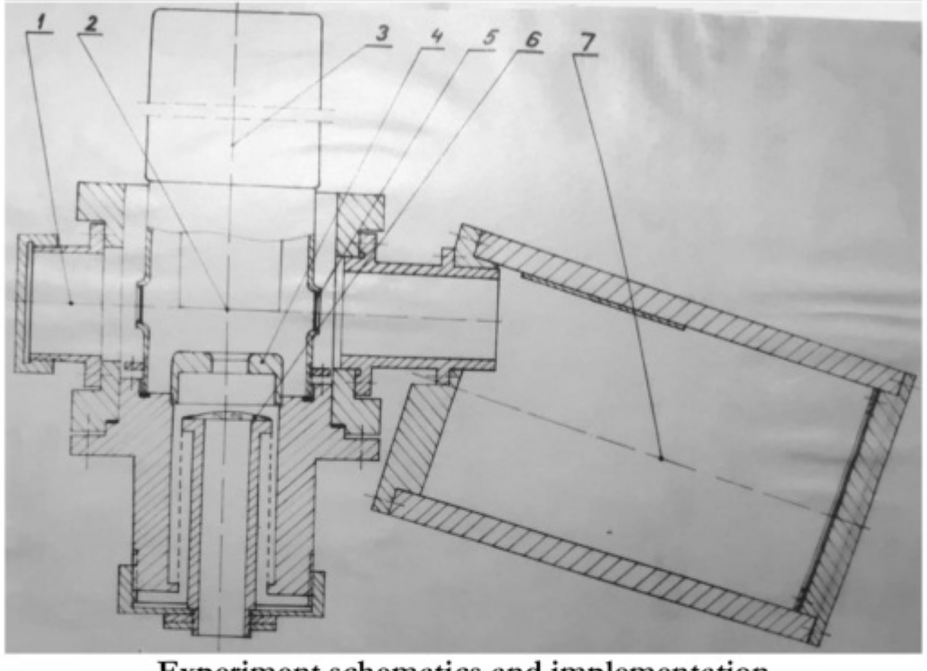
Metal engraving



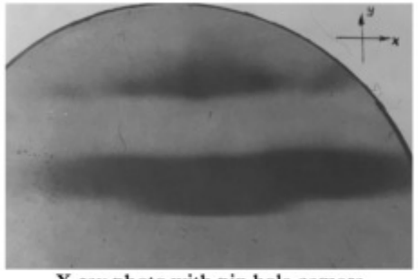
TEA CO_2 laser deposition

Sofia University, Faculty of Physics, Quantum Electronics Department (1977 – 1979)
Graduation work thesis: X-ray emission from laser plasma in a strong electric field

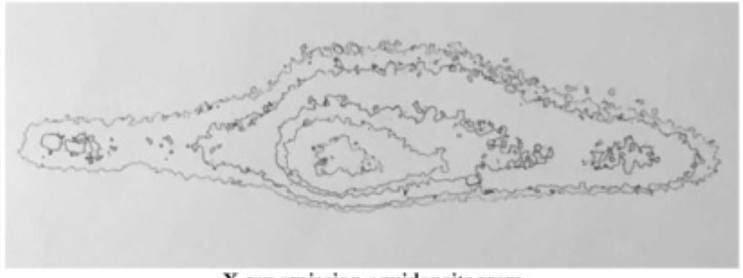




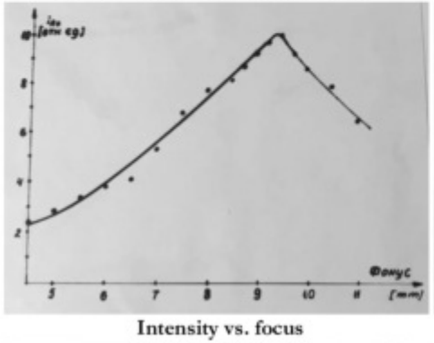
Experiment schematics and implementation



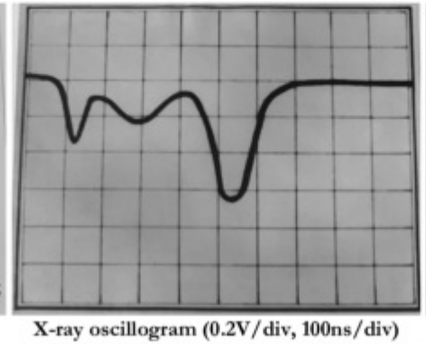
X-ray photo with pin hole camera



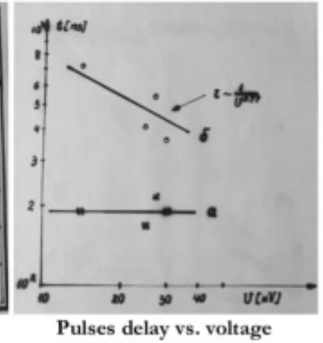
X-ray emission equidensitogram



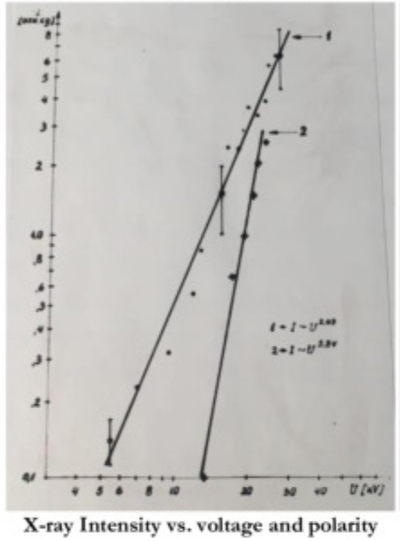
Intensity vs. focus



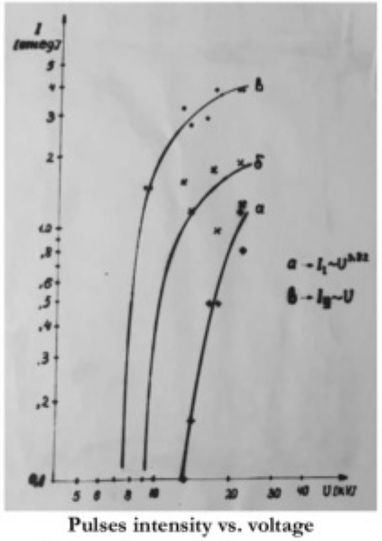
X-ray oscillogram (0.2V/div, 100ns/div)



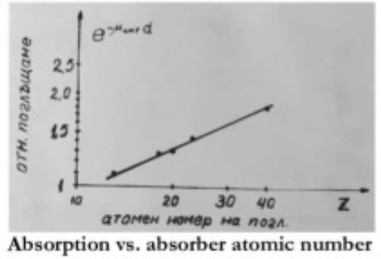
Pulses delay vs. voltage



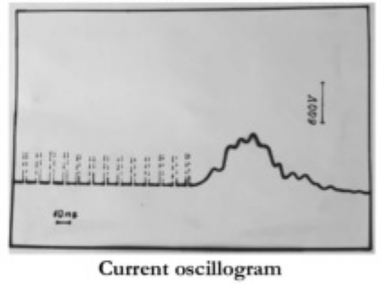
X-ray Intensity vs. voltage and polarity



Pulses intensity vs. voltage



Absorption vs. absorber atomic number



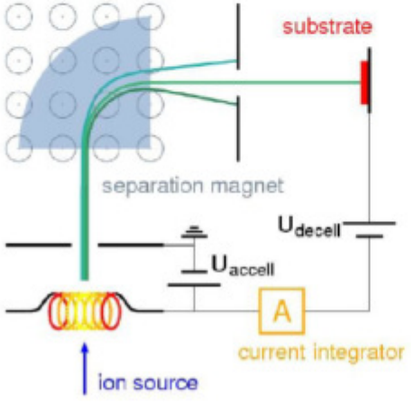
Current oscillogram

Experimental results

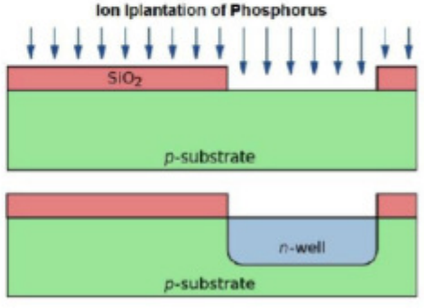
Sofia University, Faculty of Physics, Solid State Department (1974 – 1977)
Active member of semiconductor devices group

Two-dimensional photo effect in surface semiconductor transition and large area devices based thereon

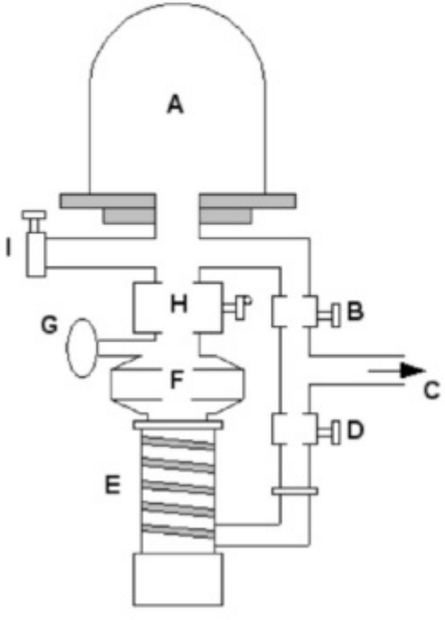
Production process of position sensitive device based on two-dimensional photo effect in surface semiconductor transition



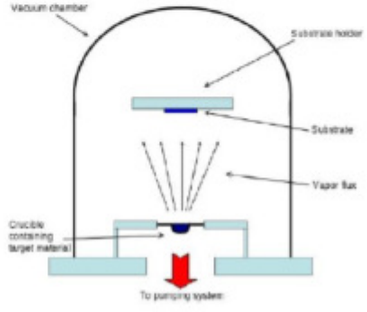
Ion Implantation of Si p-substrate



The result is thin Si disk with n-p transition on its surface

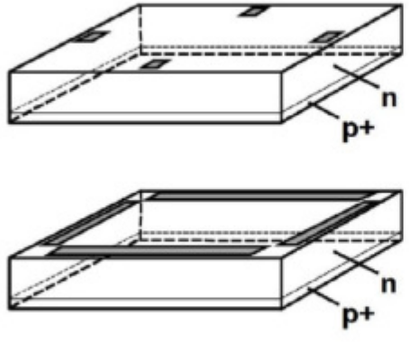


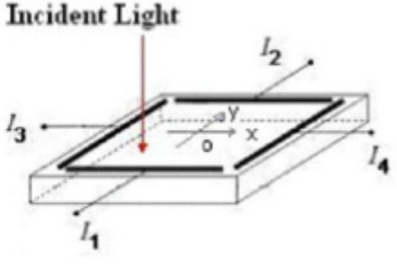
High vacuum system

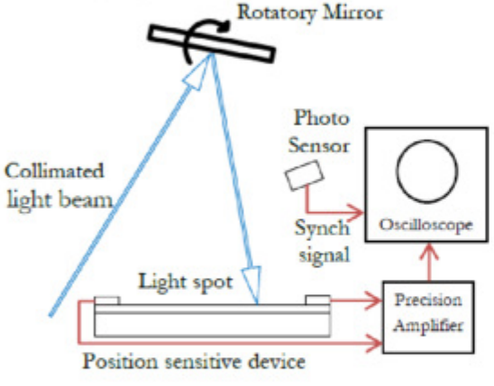


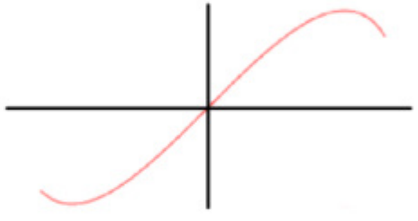
Vacuum chamber used to put thin Al layer by evaporation

Theory and proof of concept by measurement





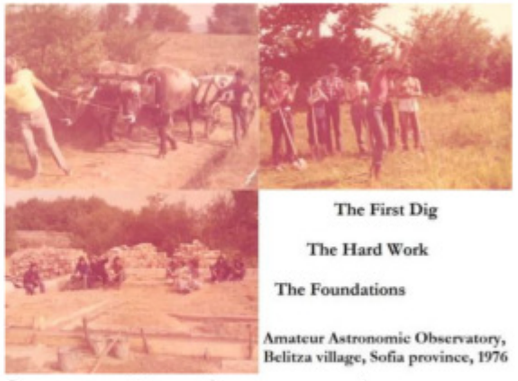




Co-founder of astronomy and optics clubs; active member of amateur radio and electronics clubs
High technical school of mechanics, Plovdiv (1968 – 1972); Voluntary organization for defence assistance (1967 – 1972);
District station of the young technicians (1968 – 1972); Sofia University, Faculty of Physics, Faculty Observatory (1974 – 1977)

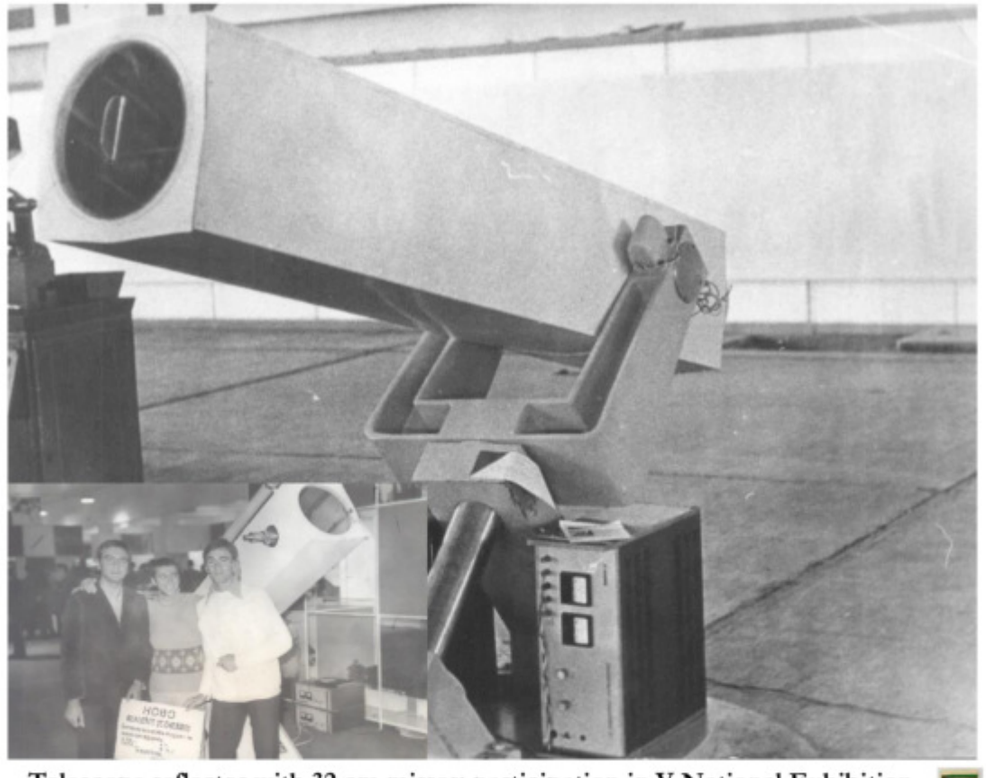
Optical club at the faculty observatory

Electronic device for image processing at the faculty observatory



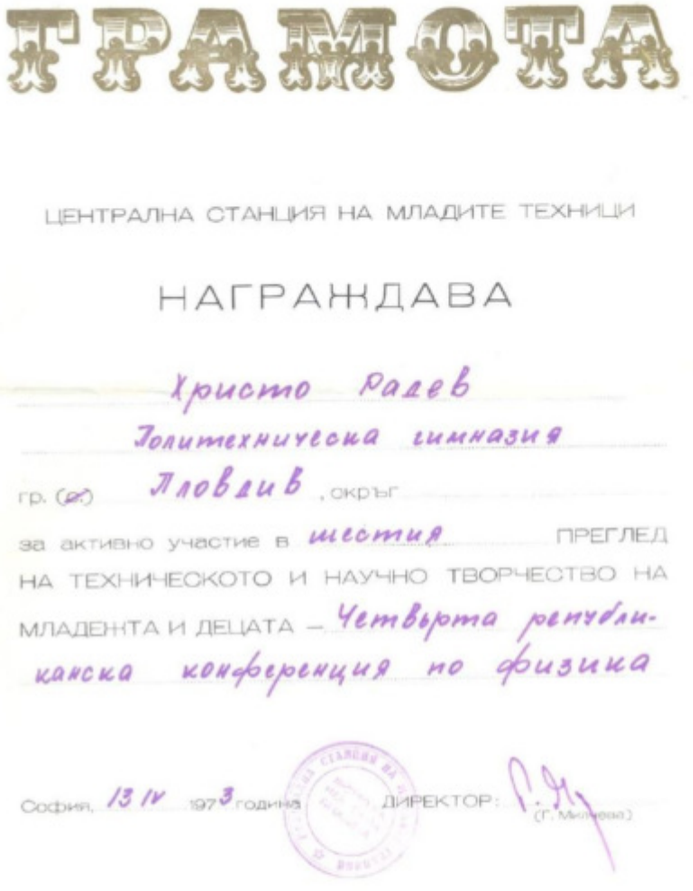
Construction of amateur observatory

Telescope reflector with 10 cm mirror: participation in IV National Exhibition for Technical and Scientific Creation of Youth, 1971, Golden medal



Telescope reflector with 32 cm mirror: participation in V National Exhibition for Technical and Scientific Creation of Youth, 1973, Golden medal

Scholar essay: Development of the philosophical categories for space and time in mathematics and physics, 1972
Participation in Mathematical school for Olympic Competitors (1968 – 1972) and IV National Conference on Physics, Sofia, 1973



ГРАМОТА

ЦЕНТРАЛНА СТАНЦИЯ НА МЛАДИТЕ ТЕХНИЦИ

НАГРАЖДАВА

Христо Радев

Политехническа гимназия

гр. (с.) Пловдив, окръг

за активно участие в шестия ПРЕГЛЕД

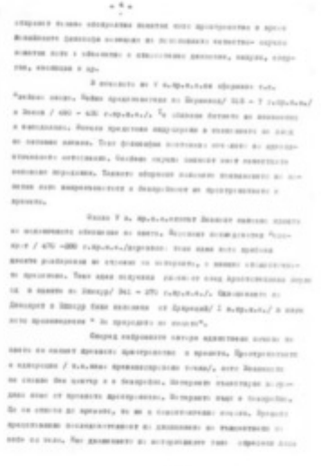
НА ТЕХНИЧЕСКОТО И НАУЧНО ТВОРЧЕСТВО НА

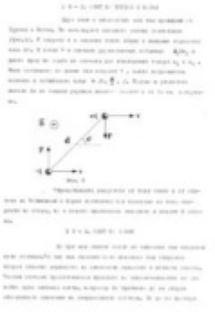
МЛАДЕЖТА И ДЕЦАТА – Четвърта републи-

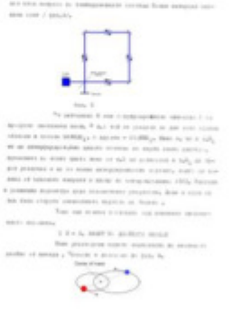
канска конференция по физика

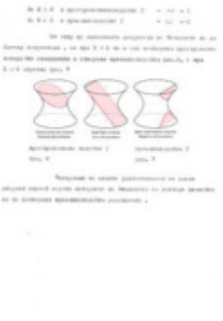
София, 13 IV 1973 година ДИРЕКТОР: (Г. Милчева)

Програма на IV Републиканска конференция по физика
Program of the IV Republican Conference in Physics

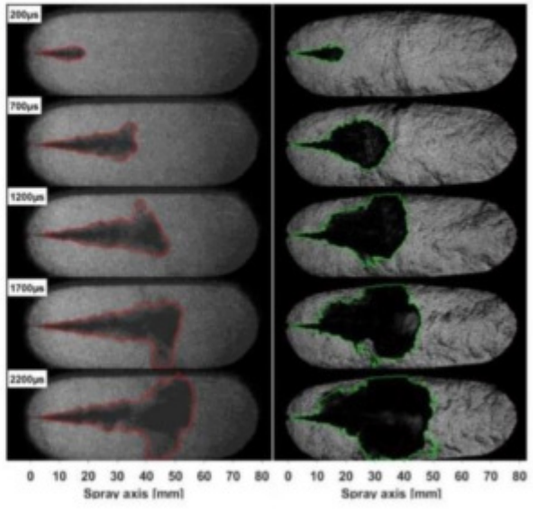




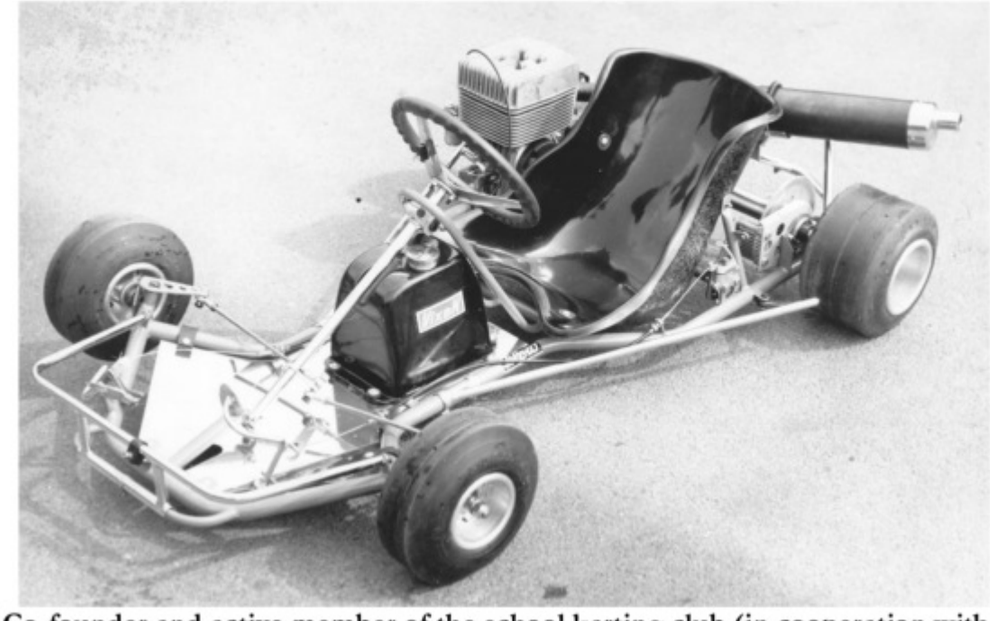




High school of mechanics, Plovdiv: Internal combustion engines speciality, School karting club (1968 – 1972)
Graduation work thesis: Investigation of the combustion processes in diesel engines



Test suit for adjustment of fuel pumps and nozzles for diesel engines - repairing and testing as a part of graduation work



Co-founder and active member of the school karting club (in cooperation with the sport cars station Bulgaralpine of Bulgarreno plant): design, crafting, race training; participation in competitions and IV National Exhibition of Technical and Scientific Creation of Youth, 1971, Silver medal for "Racing karting".