GNU/Linux and FPGA in Real-time Control Applications

Pavel Pisa
pisa@cmp.felk.cvut.cz
CC BY-SA

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Motion control hardware developed and provided by
PiKRON s.r.o.
{ppisa,porazil}@pikron.com

2017-03-05

InstallFest 2017

Content of Presentation

Parallel Hacking Task

- You can find Debian based device with open root access (ssh -p 2222 root@147.32.87.251)
- The device is Xylix Zynq-7010 running mainline 4.9 kernel with RT patches
- Root login password is B35APO
- There is 32 LEDs line peripheral mapped on the physical address 0x43c40004
- More about educational peripherals at https://cw.fel.cvut.cz/wiki/courses/b35apo/documentation/mz_apo
- Task 1: write some bit pattern on the 32 LEDs line (SPILED_REG_LED_LINE_o)
- Task 2: write some picture on the display (480× 320 565 RGB), you can find some example on the board at path /opt/apo/lcd/mzapo_lcdtest

Background Introduction

The motion control and precise positioning is fundamental for many of medical, laboratory and robotic instruments and controllers to which development I have contributed to and often coordinated at PiKRON company. I use GNU/Linux as my main host system for more than 20 years. We use it as the only host development platform at company. We use it sometimes in embedded applications in parallel to RTEMS and bare metal development.

GNU/Linux is extensively used as core network infrastructure at Department of Control Engineering and it is the main environment for all seminaries and lab works in subjects which I teach. We use Linux, RTEMS and other real-time operating systems for applications and infrastructure development done for worldwide car-makers and industrial partners as well.

Previous Presentations

- InstallFest 2015

Is Raspberry Pi Usable for Industrial and Robotic Applications?
http://installfest.cz/if15/slides/pisa_rpi.pdf

- LinuxDays 2015

Linux, RPi and other HW for DC and Brushless/PMSM Motor Control
https://www.linuxdays.cz/2015/video/Pavel_Pisa-Rizeni_stejnosmernych_motoru.pdf

- LinuxDays 2016

Processor Systems, GNU/Linux and Control Applications

Related Articles

- RTLWS 2014, OSADL, Sojka, M. – Píša, P.

Usable Simulink Embedded Coder Target for Linux
https://rtime.felk.cvut.cz/publications/public/ert_linux.pdf

- ROOT.CZ 9. 5. 2016

GNU/Linux pro řízení a rychlost jeho odezvy
https://www.root.cz/clanky/gnu-linux-pro-rizeni-a-rychlost-jeho-odezvy/

- ROOT.CZ 3. 10. 2016

Linux pro řízení: minimalistické řešení řízení stejnosměrného motoru
https://www.root.cz/clanky/linux-pro-rizeni-minimalisticke-reseni-rizeni-stejnosmerne

Related Thesis Works

The most recent theses I lead in motion control and FPGA area

- Radek Mečiar: Motor control with Raspberry Pi board and Linux, 2014 (PDF)
- Martin Meloun: FPGA Based Robotic Motion Control System, 2014 (PDF)
- Martin Prudek: Brushless motor control with Raspberry Pi board and Linux, 2015 (PDF)
- Nepivoda Tomáš: DC Motor Control Peripheral Module for Zynq Platform, 2016 (PDF)
- Martin Jeřábek: FPGA Based CAN Bus Channels Mutual Latency Tester and Evaluation, 2016 (PDF)

Electric Motors

- main categories:
  - brushed DC – motor with mechanical commutator
  - brushless DC motor (BLDC)/Permanent magnet synchronous motor PMSM – comutator replaced by control electronics – needs position/sector sensor or position estimation
  - stepper motor – synchronous motor, position usually enforced by direct drive
  - induction or asynchronous motor – exact position is not strictly required for control, torque is function of slip of rotor behind rotating magnetic field

Raspberry Pi Applications

- facts
  - intended for education
  - provides decent performance for video playback
  - is really cheap
- the last point is important
  - used for many hobby projects, strong community
  - used even in commercial solutions due to low cost even that it is not intended for such use

Alternatives

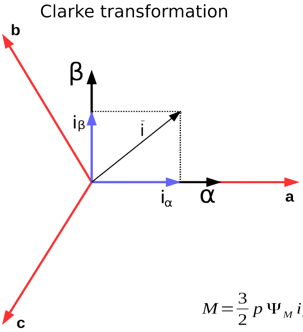
Many better alternatives exists for full featured GNU/Linux systems for industrial applications.
There are listed few ones

- FreeScale i.MX53 – ARM Cortex A8, ETHERNET, CAN, USB, ...
- FreeScale i.MX6 – ARM Cortex A9
- TI AM335x Sitara ARM Cortex-A8 (Beagle Bone black) – adds quadrature encoder inputs, two real time coprocessor units (PRU)
- Ti AM437x – Cortex A9 based, 2×PRU
- Ti AM5728 – dual core Cortex A15, PowerVR GPU, 2× Cortex M4 cores, dual core C66x DSP, IVA (H.264), 2×PRU
- Xilinx Zynq-7000 – 2×Cortex A9, FPGA

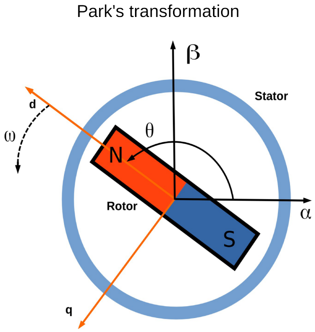
Other PowerPC, SPARC based solution are used for military and space.

Outline

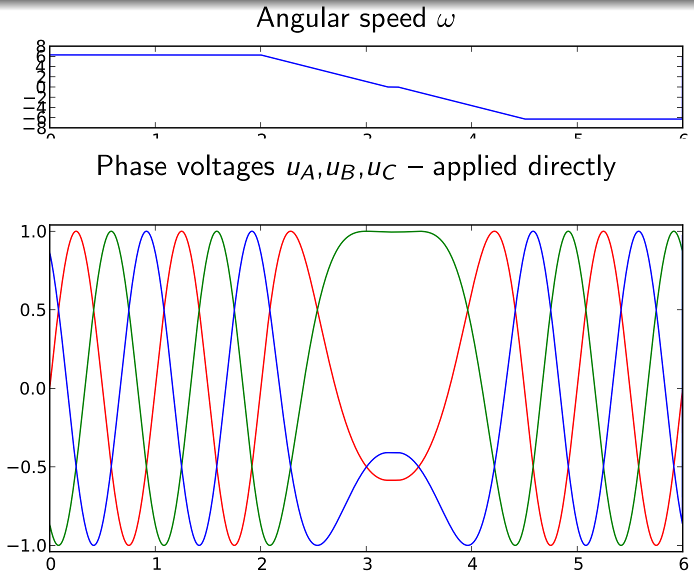
Three Phase Circuit Transformations



$$M = \frac{3}{2} p \Psi_M i_q$$



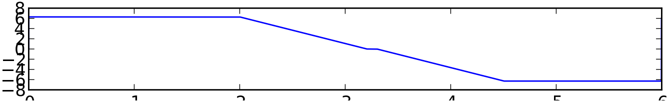
Raw Sine-wave Voltage



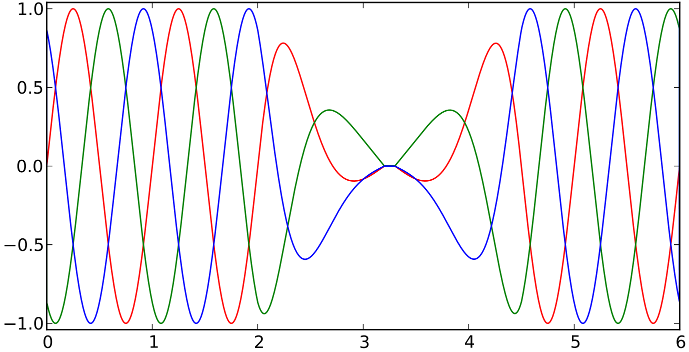
 CC BY-SA

Voltage Proportional to Speed

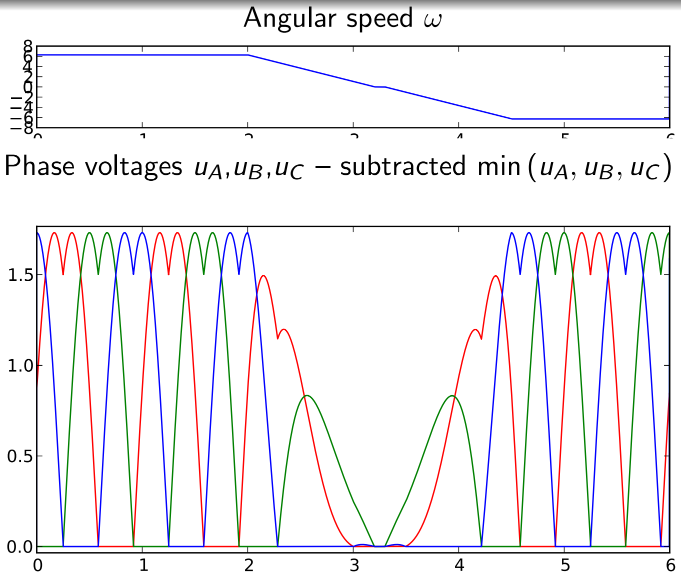
Angular speed ω



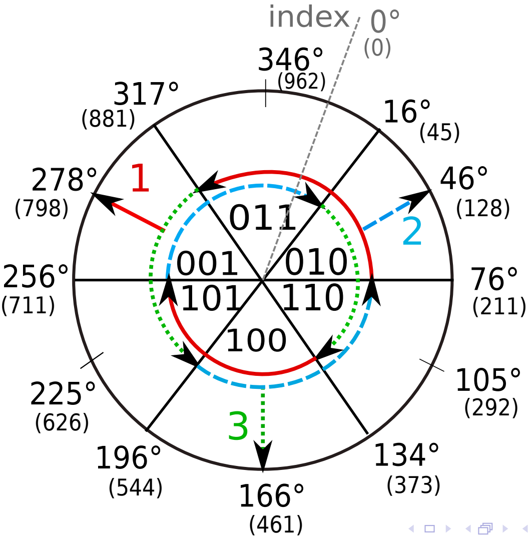
Phase voltages u_A, u_B, u_C – proportional to ω



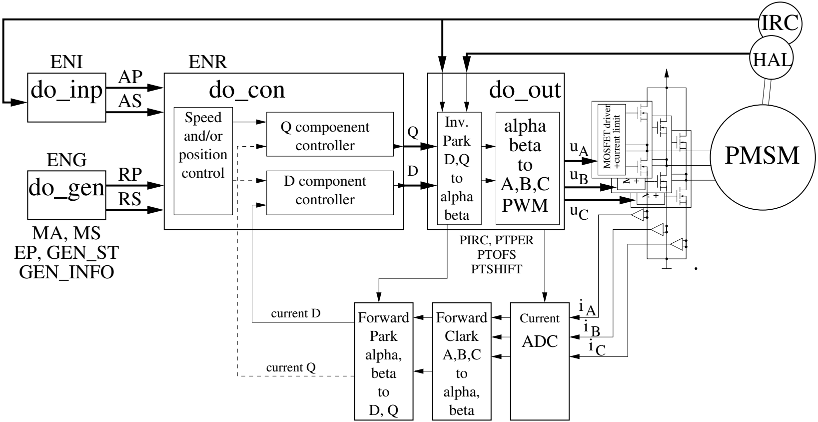
Unipolar/Zero-based Voltage



Hall Sensors to Sectored Position



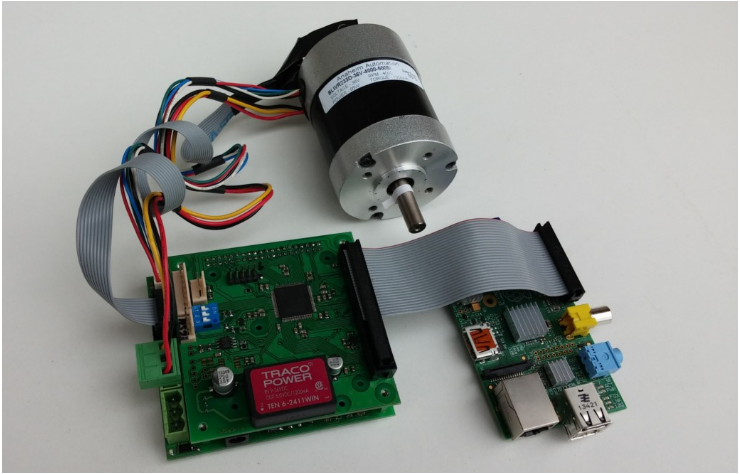
Simplified PMSM Vector Control



Source: PiKRON PXMC library

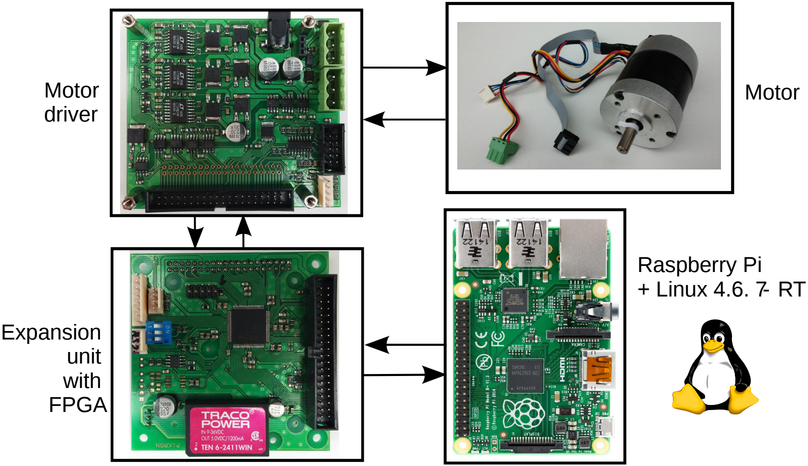
Outline

PMSM Experimental Setup

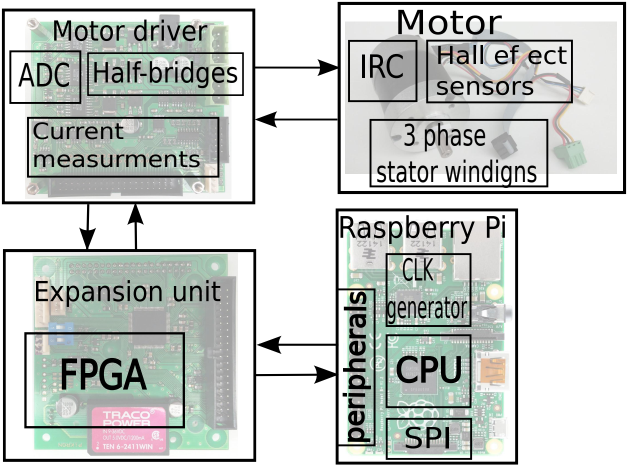


Source: Martin Prudek's Bachelor Thesis

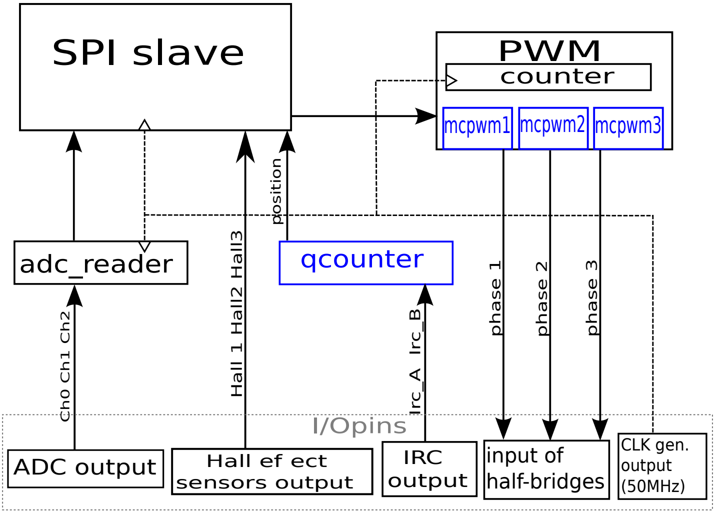
Motor, Boards and Interconnection



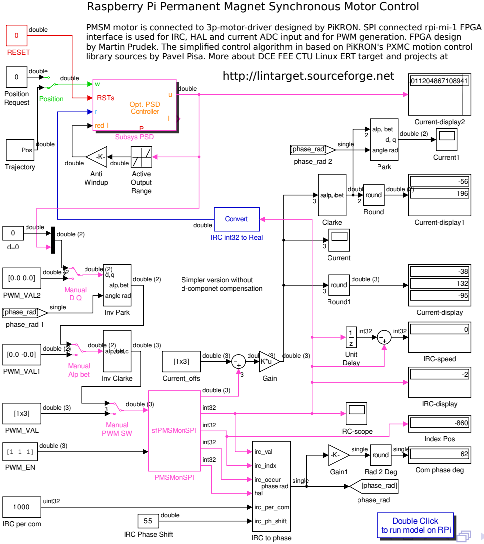
Function Placement to the Borads



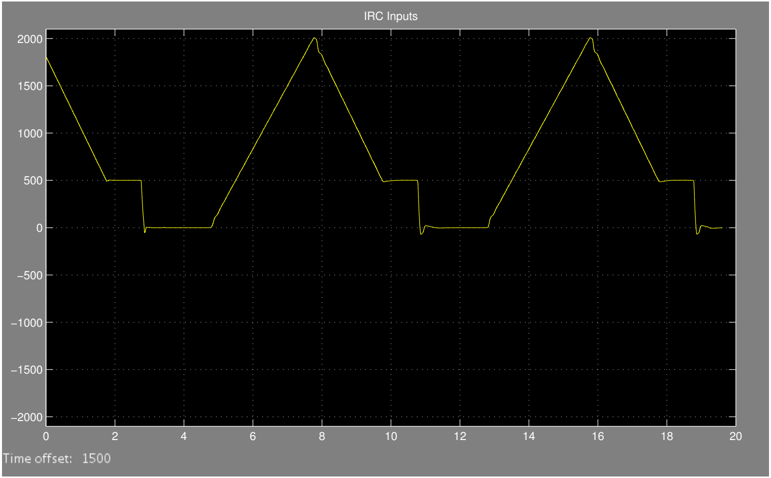
FPGA Functional Blocks



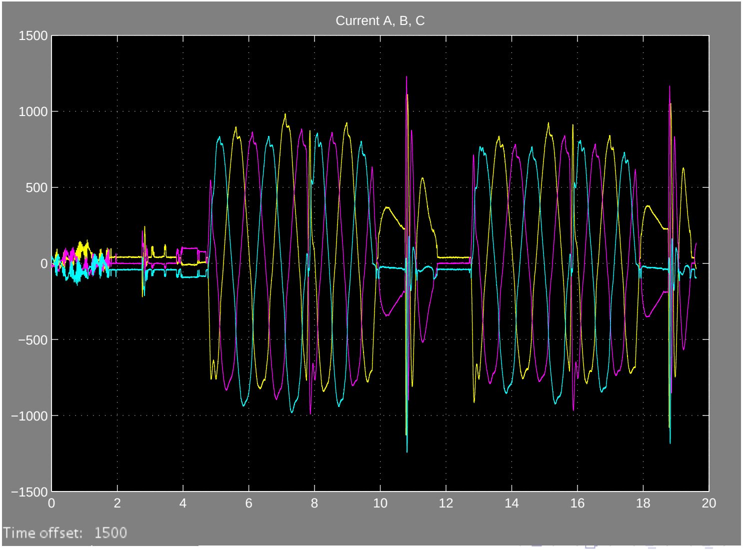
RPi PMSM Motor Control Simulink Diagram



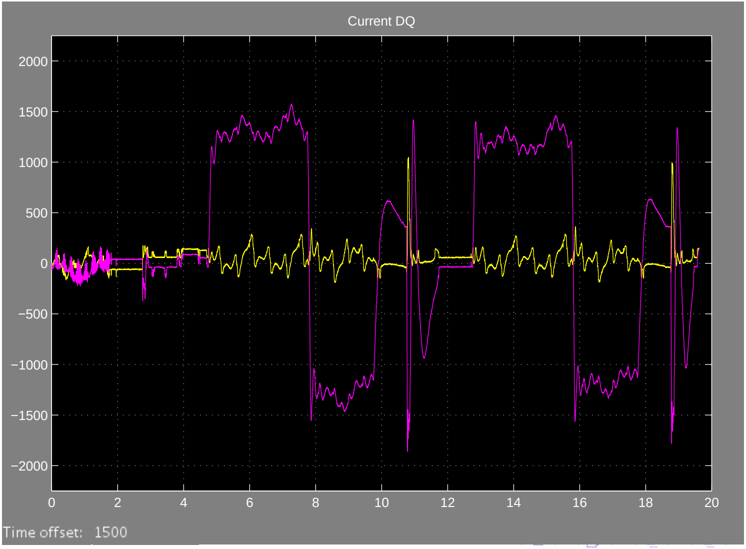
RPi PMSM IRC Plot



RPi PMSM Current A, B, C Plot

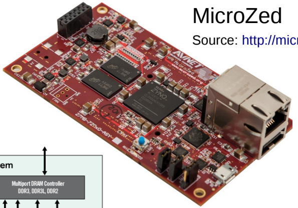


RPi PMSM IRC Current D, Q



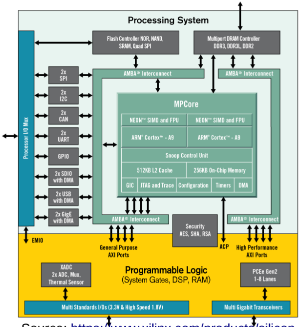
Outline

MZ_APO – Kit for Education

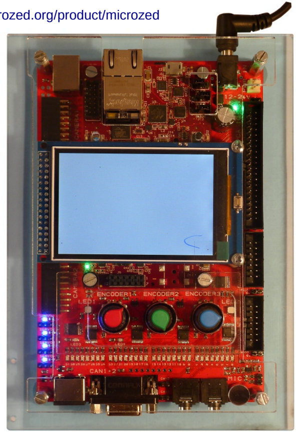


MicroZed

Source: http://microzed.org/product/microzed



Source: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html



Source: https://cw.fel.cvut.cz/wiki/courses/b35apo/start

The Core – MicroZed and System

- The core chip: Zynq-7000 All Programmable SoC
- Family member: Z-7010, device XC7Z010
- CPU: Dual ARM® Cortex™-A9 MPCore™ @ 866 MHz (NEON™ & Single / Double Precision Floating Point) 2× L1 32+32 kB, L2 512 KB
- FPGA: 28K Logic Cells (˜430K ASIC logic gates, 35 kbit)
- Computational capability of FPGA DSP blocks: 100 GMACs
- Memory for FPGA design: 240 KB
- Memory on MicroZed board: 1GB
- Operating system: GNU/Linux
    - GNU LIBC (libc6) 2.19-18+deb8u7
    - Kernel: Linux 4.9.9-rt6-00002-ge6c7d1c
    - Distribution: Debian Jessie

MZ_APO Hardware Education Kit

- The kit is designed to be universal for multiple couses
  - Computer architectures courses (B3B35APO, B4B35APO)
  - Advanced Computer Architectures (B4M35PAP, BE4M35PAP)
  - Real -Time Systems Programming (B3M35PSR)
- Interfaces accessible directly on MicroZed module (single board computer SBC)
  - 1G ETHERNET
  - USB Host, connector A
  - serial port UART1 connected to USB to serial converter soldered on the module, USB micro-B
  - micro SD card slot
  - on-board Flash,
  - one user controlled LED
  - reset switch and user switch

MZ_APO Board Peripherals

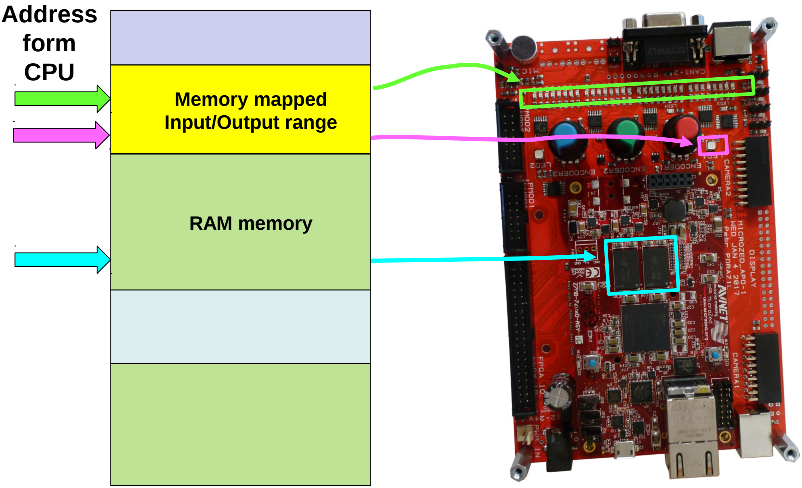
- Small 16-bit parallel bus connected LCD display (480 × 320, RGB 565)
- 32 LEDs for direct visualization of 32-bit word (SPI connected), presented as single 32-bit register in physical memory address space
- 2× RGB LED (SPI connected, 8-bit PWM), 2×32-bit register
- 3× incremental encoder rotary knob (RGB 888, SPI connected) , presented as 3 three 8-bit fields in 32-bit word
- 4× RC model servo interface (5 VDC power and pulse output, resolution 10 ns)
- 1× 40 pin FPGA IO connector, 36 FPGA 3.3 VDC signals, jumper enables +5 VDC power, signals match Altera DE2 kits which we use in other courses
- 2× PMOD connectors extended by optional +5 VDC power, each provides 8 FPGA signals shared with FPGA IO connector

MZ_APO Board Peripherals (cont.)

- 2× parallel camera interface, one 10-bit and one 8-bit
- 2× CAN bus transceivers, 5 Mbit/s capable, connected to Xilinx CAN peripheral, but FPGA CAN-FD possible in the future
- audio output by simple PWM modulator, on-board speaker and JACK available
- audio input to Xilinx integrated ADC, on-board microphone and JACK
- standard notebook power supply JACK 12 to 24 VDC
- UART0 serial port, galvanic isolated connection to FTDI serial to USB chip, robust USB A connector
- break signal longer than 1s resets the board

Full list at `https://cw.fel.cvut.cz/wiki/courses/b35apo/documentation/mz_apo/start`

Peripherals Mapped into Physical Address Space



Source: ČVUT FEL Computer Architectures Course https://cw.fel.cvut.cz/wiki/courses/b35apo/start

PMSM/BLDC Peripheral registers

Peripheral registers block at Z3PMDRV1_REG_BASE_PHYS 0x43c20000

Register	Offset	Description
Z3PMDRV1_REG_IRC_POS_o	0x08	32-bit position
Z3PMDRV1_REG_IRC_IDX_POS_o	0x0C	position of index
Z3PMDRV1_REG_PWM1_o	0x10	14-bit PWM
Z3PMDRV1_REG_PWM2_o	0x14	bit 30 enable
Z3PMDRV1_REG_PWM3_o	0x18	bit 31 shutdown
Z3PMDRV1_REG_ADC_SQN_STAT_o	0x20	HAL and status bits
Z3PMDRV1_REG_ADC1_o	0x24	24 - bit ADC
Z3PMDRV1_REG_ADC2_o	0x28	cumulative
Z3PMDRV1_REG_ADC3_o	0x2C	sums

HAL and Status Register Bits

Symbol	Mask	Description
Z3PMDRV1_REG_ADSQST_SQN_m	0x000001FF	ADC cycles
Z3PMDRV1_REG_ADSQST_HAL1_m	0x00010000	HAL1
Z3PMDRV1_REG_ADSQST_HAL2_m	0x00020000	HAL2
Z3PMDRV1_REG_ADSQST_HAL3_m	0x00040000	HAL3
Z3PMDRV1_REG_ADSQST_ST1_m	0x00100000	Overload/ error signal phase driver
Z3PMDRV1_REG_ADSQST_ST2_m	0x00200000	
Z3PMDRV1_REG_ADSQST_ST3_m	0x00400000	
Z3PMDRV1_REG_ADSQST_PWST_m	0x01000000	power present

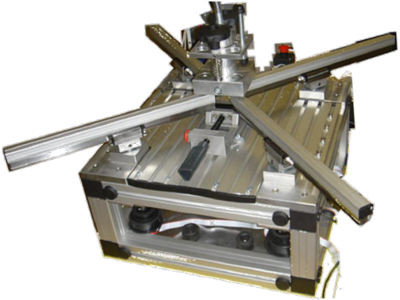
Actual Code Sources

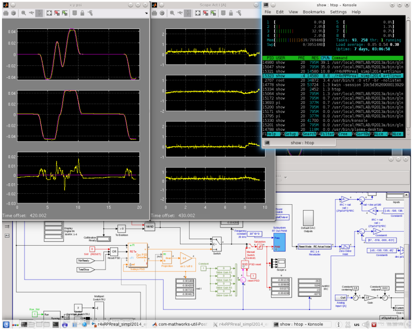
- VHDL sources of SPI connected PMSM peripheral for Raspberry Pi
  https://rtime.felk.cvut.cz/gitweb/fpga/rpi-motor-control.git/tree/HEAD:/pmsm-control
- Repository with experimental sources with C based motion control code port, only for testing, locking and synchronization not complete for SMP systems
  https://rtime.felk.cvut.cz/gitweb/fpga/rpi-motor-control-pxmc.git/blob/HEAD:/src/app/rpi-pmsm-test1/zynq_3pmdrv1_mc.c
- Matlab/Simulink Lintaget pages
  http://lintarget.sourceforge.net/
- Matlab/Simulink model for Raspberry Pi and Xilinx Zynq
  https://github.com/ppisa/rpi-rt-control/tree/master/simulink

SocketCAN Simulink Blockset

- The blockset is quick proof port of the CAN Autosar API based blocks developed at DCE initially for own automotive grade ARM Cortex-R4 based embedded platform
- The code is generated under designed control of TLC (Target Language Compiler) blocks description which allows to optimize blocks code for used data-types and interconnection
- For more information about embedded systems rapid prototyping support developed in our group look at `http://rtime.felk.cvut.cz/rpp-tms570/`
- Notices about more Linux and embedded hardware used, tested and even some designed look at `https://rtime.felk.cvut.cz/hw/`

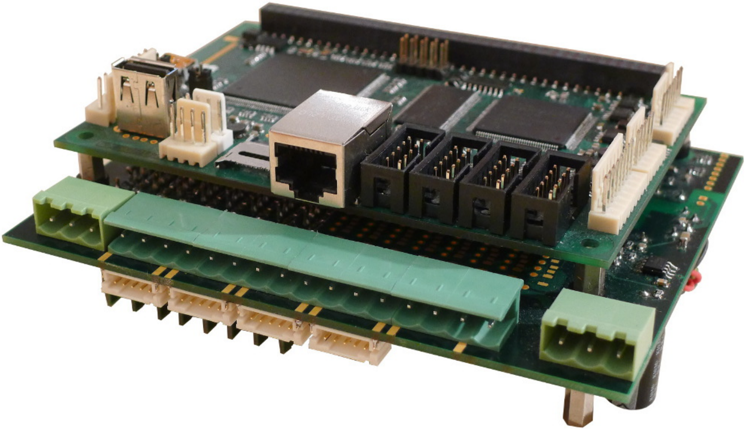
x86 Linux ERT and Parallel Kinematic Robot Control





- 4 DC motors, 4 incremental encoders, other I/Os
- Presented at Embedded world 2014
- Sampling period 1 ms but complex computations
- More reliable that previously used Windows target

LX_ROCON – DC, BLDC/PMSM and Stepper Controller

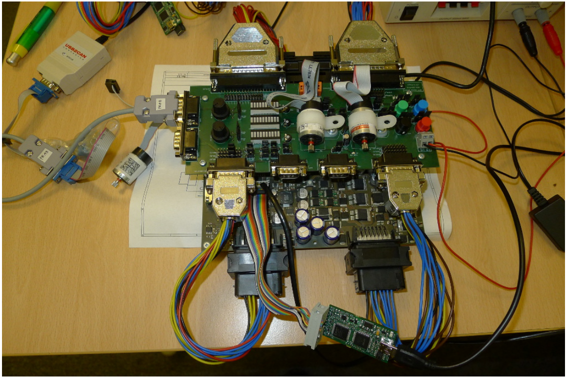


Source: PiKRON

LX_ROCON – Features

- Industry-proven Electric Motor Control libraries and system
- FPGA-based Based on Cortex-4 MCU and Xilinx Spartan 6 FPGA
- Fully configurable, 16 power outputs can be assigned up to 4× BLDC/PMSM, stepper motors and or up to 8× DC motor
- FPGA design with inferred blocks only
- Portable to MicroSemi FPGAs, does not use vendor-specific blocks
- CAN, ETHERNET, Serial, RS-485 and USB communication
- RTEMS, Nuttx and mbed supported

Cortex-R4 Automotive Platform and Test Board



TMS570LS3137 (EP) N2HET, no IRC specific peripheral

TMS570 Motion Control for Safety Applications

- TMS570LC4357 / RM57L843 ARM Cortex-R5F, two cores in lockstep, 4 MB Flash, 512 kB SRAM, 3 kB I, 3 kB D Cache, all memory with ECC, 2× QEP, N2HET 64
- TMS570LS1224 / RM46L840 ARM Cortex-R4F, lockstep, 1280 kB Flash, 192 kB SRAM, no cache, all memory with ECC, 2× QEP, N2HET 44
- No MMU, not reasonable target for Linux, when SDRAM used then it is without ECC and SDRAM is not well suitable for critical applications
- Freescale/NXP alternative PowerPC based systems MPC57xx (eTPU)
- Ti tools with FreeRTOS, more or less demo for proprietary SAFERTOS, other uC/OS-II, CoDeSys, SCIOPTA
- RTEMS BSP development started in GSoC frame, support includes chip initialization and lwIP based networking (not fully integrated yet)

Industrial Motion Control Options

- DSP Ti C2000, Freescale/NXP MC56F84XXX, MC56F82xxx
- MCUs Ti TM4C ARM Cortex-M4F, STM32 Nucleo+IHM02A1, Freescale/NXP Kinetis V ARM Cortex-M0+/M4/M7
- GNU/Linux capable systems Ti AM335x, Ti AM437x

Conclusion

- More ready to be uses open-source building blocks for control applications have been presented and are available online
- We are looking for students who has interest in real-time, operating systems and control/embedded hardware
- We cooperate with more industrial partners on many projects and students can gain experience and valuable knowledge during their work on the project in frame of thesis
- We offer control related courses Real-Time systems programming and participate on generic computer architectures courses at CTU FEE

Thanks for attention and questions