PHILIPS

sense and simplicity

Electronics & SW Architecture 1v1

Sait Izmit

Philips Applied Technologies

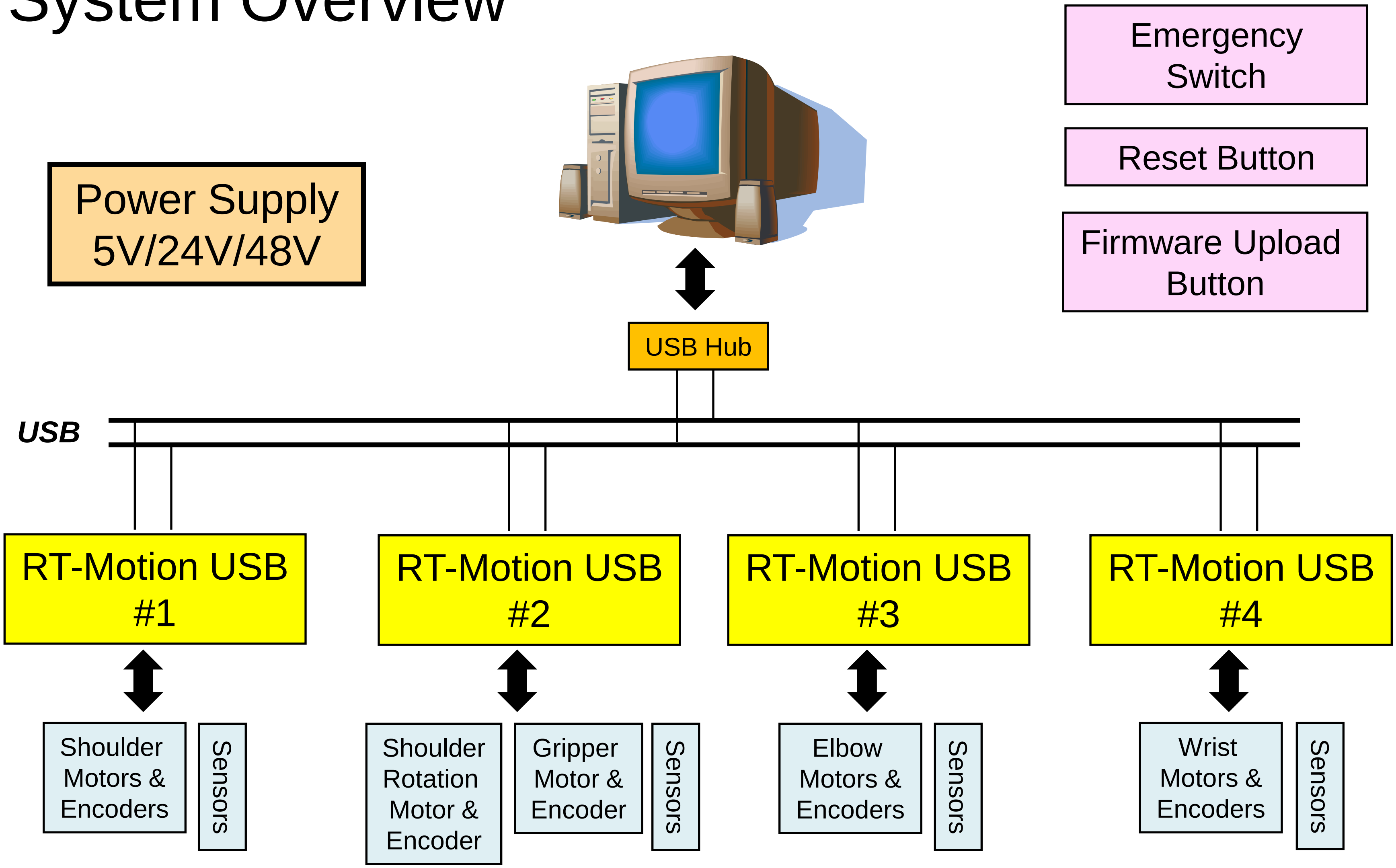
Philips Experimental Robot Arm

December 9, 2009

Agenda

- System Overview
- RT-Motion USB
- PC-Side Software
- Recommended Use
- Software Alternatives

System Overview



Power Supply
5V/24V/48V

Emergency Switch

Reset Button

Firmware Upload Button

USB Hub

USB

RT-Motion USB #1

RT-Motion USB #2

RT-Motion USB #3

RT-Motion USB #4

Shoulder Motors & Encoders

Sensors

Shoulder Rotation Motor & Encoder

Gripper Motor & Encoder

Sensors

Elbow Motors & Encoders

Sensors

Wrist Motors & Encoders

Sensors

System Configuration

- Detailed Wiring Diagram will be supplied with the final documentation

- Each RT-Motion USB board is specially modified for the defined position. Therefore,
  - Boards shouldn’t be swapped with each other
  - Motor wirings shouldn’t be modified

- Other small PCBs are used due to ease of cabling, power limits of cables/components etc. DON’T MESS WITH THE CABLING

Agenda

- System Overview
- RT-Motion USB
- PC-Side Software
- Recommended Use
- Software Alternatives

PHILIPS

RT-Motion USB: What is it?

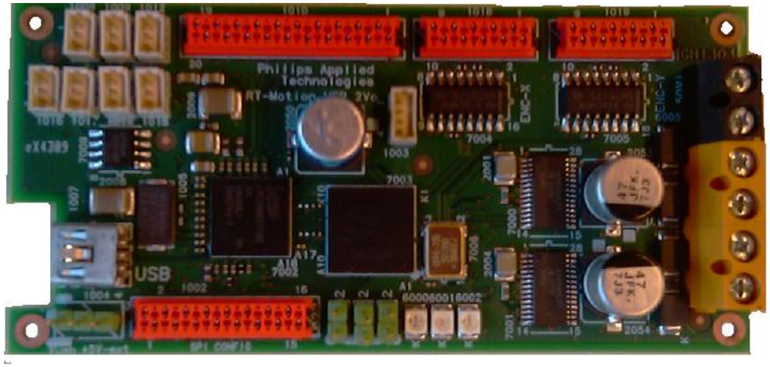
- Complete 2-axes motion control solution with integrated drive units & extensive I/O
- Used for high-speed USB-based distributed motion control applications
- Supported by a real-time software architecture
- Some specs:
  - 32-bit Processor
  - 16 bit Digital I/O (3.3V – 5V input tolerant)
  - 1x10-bit 5ch Analog Inputs (0-3V), 2x16-bit Analog Inputs (0-3.3V)
  - 2x16-bit Analog Output (0-2.7V) (used to drive amplifiers)
  - FPGA-based HW Encoder Counter
  - External- or USB- Powered
  - 2 x 150Watt DC Motor Amplifiers

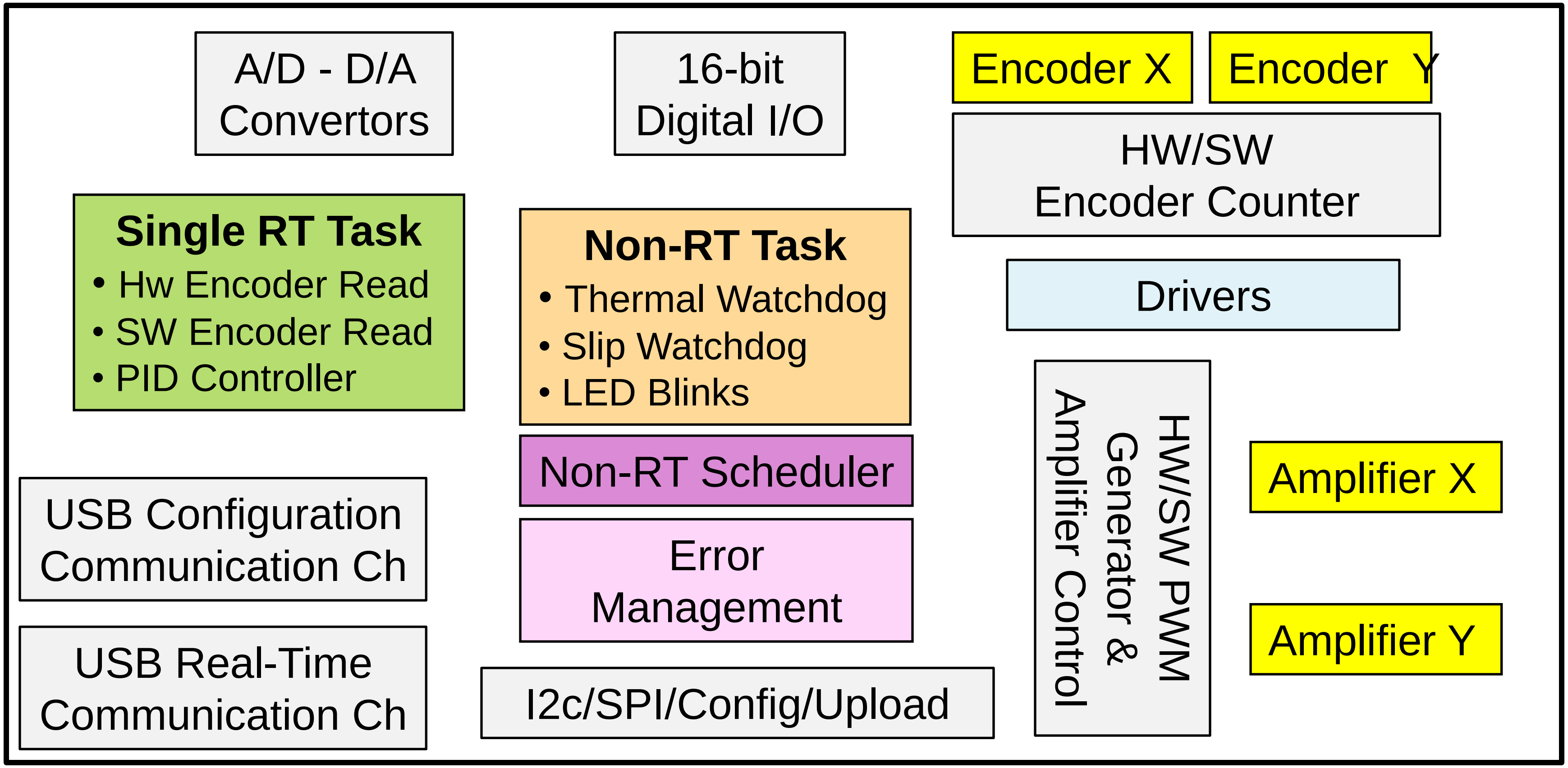
RT-Motion USB: Licensing

- Proprietary of Philips Electronics
- All software only to be used with RT-Motion USB boards
- RT-Motion USB board only to be used with Philips Applied Technologies' firmware
- Firmware may not be re-distributed. Do not put on external or internal web!
- RT-Motion USB boards and SW is provided for use with the robot arm only

PHILIPS

RT-Motion USB: Functional Overview





A/D - D/A Convertors

16-bit Digital I/O

Encoder X

Encoder Y

HW/SW Encoder Counter

Single RT Task

- Hw Encoder Read
- SW Encoder Read
- PID Controller

Non-RT Task

- Thermal Watchdog
- Slip Watchdog
- LED Blinks

Drivers

Non-RT Scheduler

HW/SW PWM Generator & Amplifier Control

Amplifier X

Amplifier Y

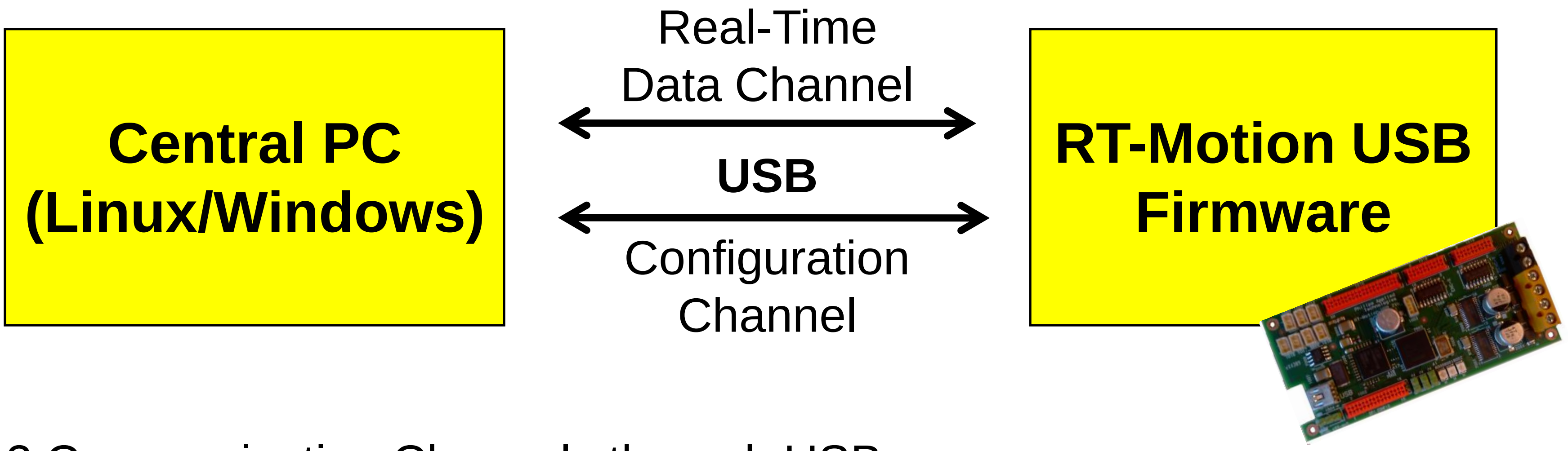
USB Configuration Communication Ch

Error Management

USB Real-Time Communication Ch

I2c/SPI/Config/Upload

RT-Motion USB: PC Communication



Central PC (Linux/Windows)

Real-Time Data Channel

USB

Configuration Channel

RT-Motion USB Firmware

- 2 Communication Channels through USB
  - Real-Time Data Channel
  - Configuration Channel
- Executed within Interrupt Service Routine context
- Real-time data write (PC to Firmware) also fills the buffers for next read
- Message indexing is possible for data verification

RT-Motion USB: LEDs

- **Green LED:** Green after initialization, blinking whenever non-real-time scheduler is running
- **Orange LED:** Indicates how much processing power the hard real-time task is using. It should never be fully on. There should be enough processor bandwidth for USB data interrupts and non-real-time tasks
- **Red LED:** Indicates firmware error (see next slide)

RT-Motion USB: Firmware Errors

- 4 Errors States:
  - No Error
  - Error – Error Code logged and Red LED is on
  - Halt Error – Error Code Logged, System Halted (all outputs disabled), Red LED blinking
  - Fatal Error – Firmware terminated

- When Halt Error occurs, error needs to be cleared for output access
- Using user API, error count, error state and the last 4 error codes can be read back
- Thermal watchdog and slip coupling watchdog will generate halt errors

RT-Motion USB: Thermal Watchdog

- Maximum current on all motors are limited in hardware on the RT-Motion USB boards
- However the elbow and wrist motors are allowed to go above the maximum continuous current limit of the motor for a short amount of time
- In order to make sure that the thermal time constant of the motor is not exceeded, a non-RT task is implemented in order to monitor RMS motor current against time
- Halt error is generated if case of (estimated) thermal overload
- Higher level user application should also do this check. **Since this is a non-RT task, do not rely on it fully!**

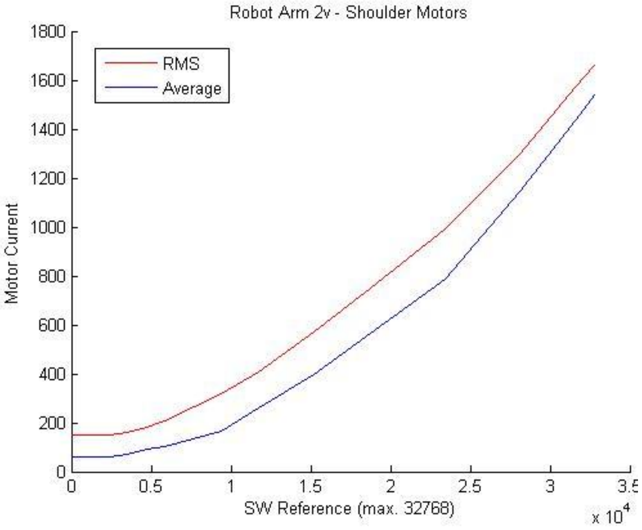
RT-Motion USB: Slip Coupling Watchdog

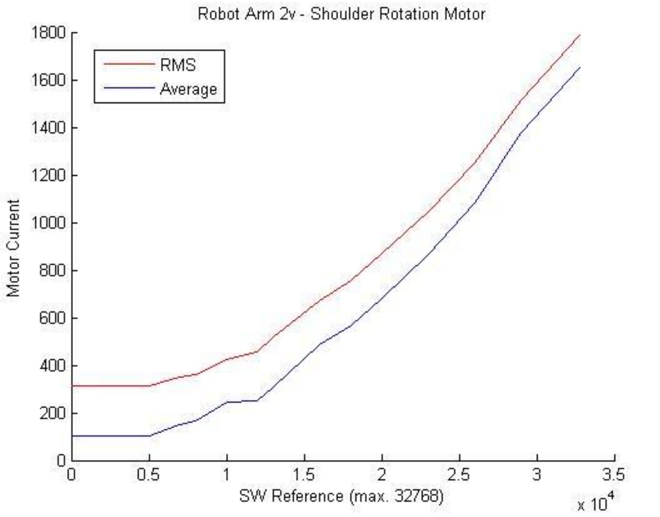
- Slip coupling on the shoulder
- 2 shoulder relative encoders (on the motor) and 2 shoulder absolute encoders can be used to calculate slip
- A non-RT task is implemented in order to monitor slip
- If slip is bigger than the limit, halt error is generated
- Higher level user application should also do this check. **Since this is a non-RT task, do not rely on it fully!**

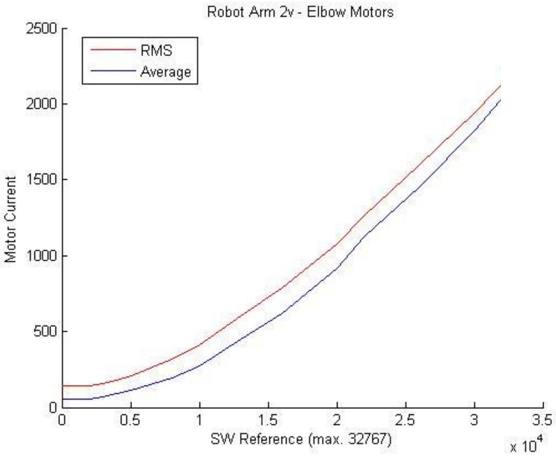
RT-Motion USB: New Firmware Upload

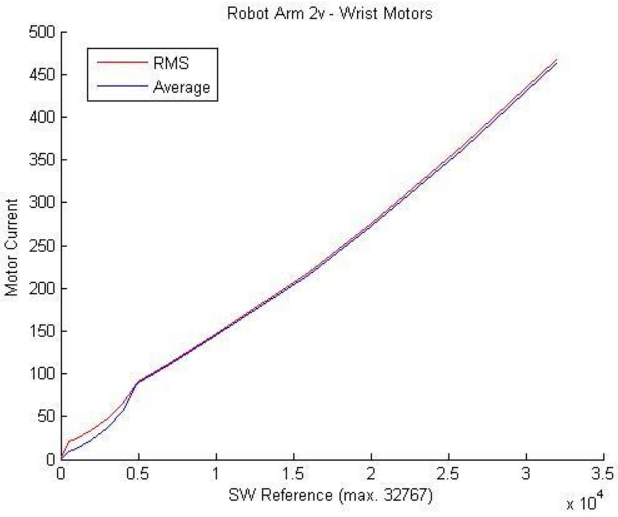
- Firmware updates can be performed by Apptech personnel and tooling
- Alternatively new firmware can be uploaded to RT-Motion USB boards via USB (“As Is” not guaranteed, 3rd party SW)
  - Procedure (See documentation for details):
    - Turn-off power and set upload switch on
    - Turn-on power
    - Start DFU application on Windows PC, go to File→ Setup
    - Select firmware binary file, click new topology
    - Connect USB cable to Windows PC (install drivers if asked)
    - After 4 boards appear in topology, File→Program
    - Click “Start Automatic Programming”
    - When finished, turn-off power, switch off program switch
    - Turn-on power
- FPGA image can only be changed by Apptech personnel and tooling

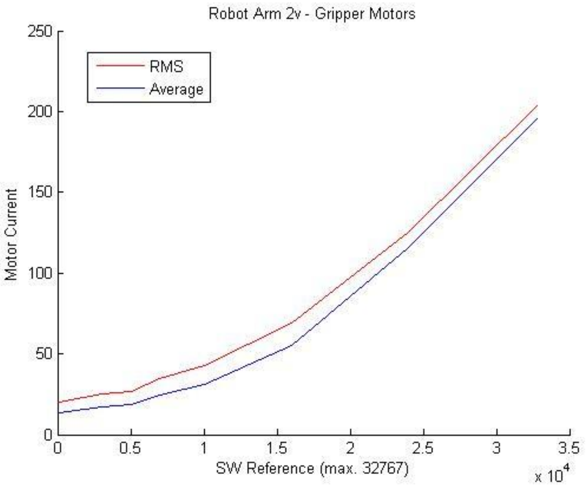
RT-Motion USB Amplifier Tuning for Robot Arm











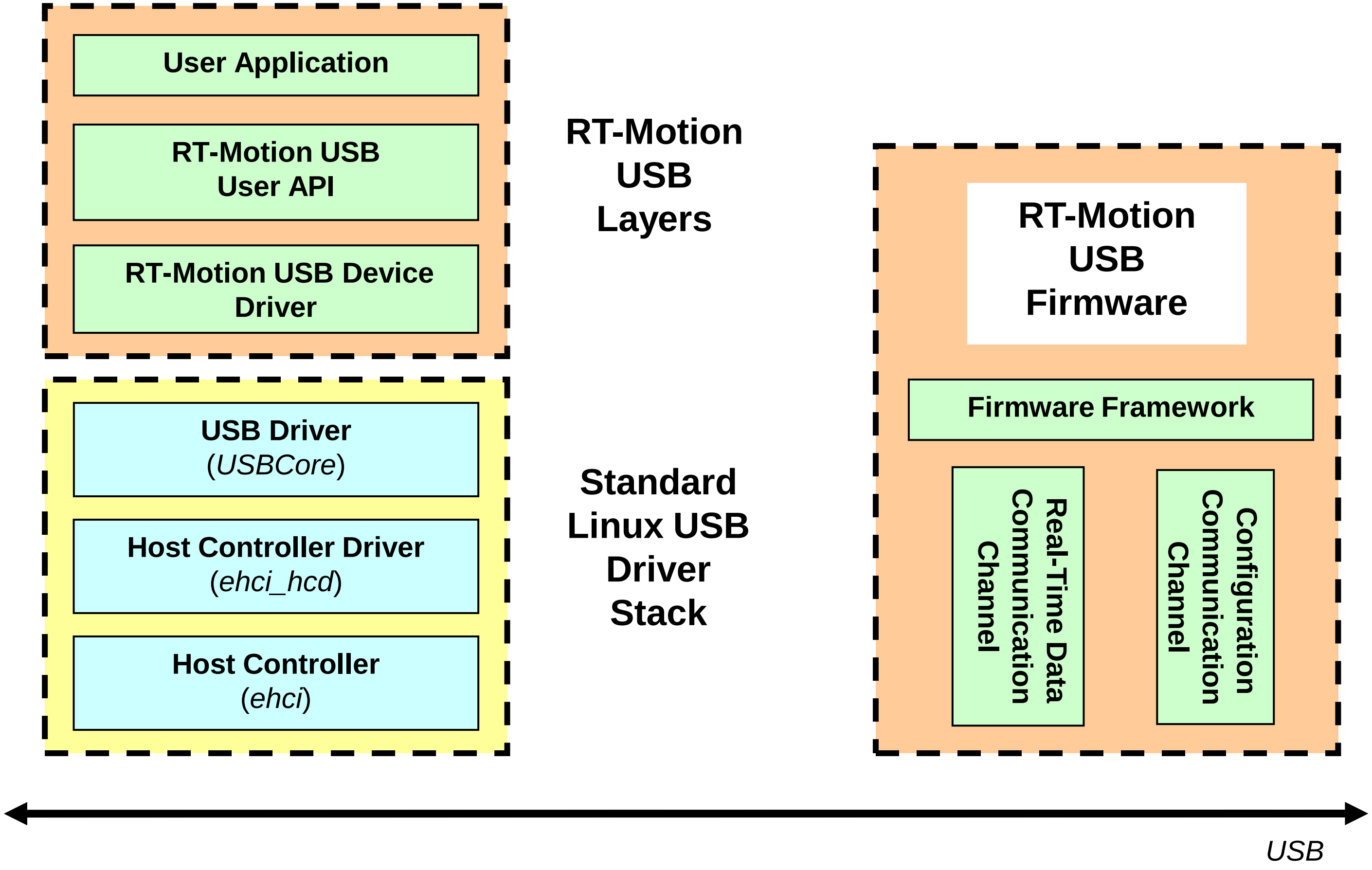
Agenda

- System Overview
- RT-Motion USB
- PC-Side Software
- Recommended Use
- Software Alternatives

Linux & Xenomai

- Xenomai is a real time co-kernel which cooperate with Linux via Adeos
- Xenomai enables the implementation of hard real time applications
- Linux (Ubuntu) in cooperation with Xenomai is used as the real-time PC platform for the robot arm application
- RT-Motion USB device driver is a native Linux driver running in kernel space
- Motion control applications should be implemented as a real-time Xenomai task in the user space

Architectural Overview under Linux


User Application
RT-Motion USB User API
RT-Motion USB Device Driver
RT-Motion USB Layers
USB Driver (*USBCore*)
Host Controller Driver (*ehci_hcd*)
Host Controller (*ehci*)
Standard Linux USB Driver Stack
RT-Motion USB Firmware
Firmware Framework
Real-Time Data Communication Channel
Configuration Communication Channel
USB

Licensing

- Device Driver is open source software and it is executed as native Linux Driver in kernel space
- User API is proprietary software of Philips and it is not allowed to re-distribute it
- **STUDY THE LICENSING STATEMENTS CAREFULLY BEFORE USING THE SOFTWARE** (They can be found in documentation or software)

RT-Motion USB User API

- Provides the firmware configuration functions and the real-time data communication functions

- Provided as a binary file

- A detailed documentation which explains user API functions and their use will be provided to you

Example User Application

Demo

Agenda

- System Overview
- RT-Motion USB
- PC-Side Software
- Recommended Use
- Software Alternatives

Sampling Time Choice

- The setups, which will be delivered to you is configured for a sampling frequency of 500 Hz
- You should make sure that the processor has enough bandwidth to run the controllers without over running previous sampling instances
- Processor bandwidth should be checked in 2 levels
  - Xenomai commands can be used to see how much processing bandwidth the real-time user application is using (Ubuntu system monitor will not show this)
  - However since the USB driver is in the Linux space, it must be make sure that Linux space has enough processing bandwidth (you can check this using Ubuntu System Monitor)

RT-Motion USB Amplifier Tuning

- Under no circumstances change the amplifier configurations, **you risk serious damage and safety**
- Input-output response is not perfectly linear but sufficient for use.
- If required, user can implement a look-up table in the user application to compensate for amplifier non-linearity

Recovery from Firmware Error State

- Whenever an error state occurs, the user can read back error state, error count and the last four error codes
- Even though the user can clear errors, especially after halt errors, it is advised to reset the firmware in order to avoid firmware instability
- It is not guaranteed after error clear the firmware is still stable

Maximum Shoulder Speed

- The robot arm should never be used beyond the speed limitations defined in the specs
- Especially when the shoulder is moving downwards with the whole weight of the arm, it is physically possible that the shoulder motors can rotate faster than the maximum defined motor speed
- If max. motor speed is exceeded, the motor will generate more voltage than the power supply
- Over voltage may destroy RT-Motion USB boards
- Clipping circuitry will be introduced to the power supply for protection but **STAY WITHIN THE SPECS!!!**

Thermal & Slip Coupling Watchdog Usage

- Higher level user application should also do these checks
- Since these watchdogs are implemented as non-real-time tasks, **DO NOT FULLY RELY ON FIRMWARE FOR PROTECTION!**

USB Usage

- Do not connect any other USB device (incl. keyboard and mouse) to the Linux/Xenomai PC
- External USB devices may effect real-time behavior
- Use the keyboard and mouse provided by us

Agenda

- System Overview
- RT-Motion USB
- PC-Side Software
- Recommended Use

Windows Driver

- Provided as binary (dll)
- Experimental driver
- Proprietary software

Matlab Toolbox

- Experimental toolbox which calls functions from RT-Motion USB Windows library
- Has the same function names and parameters as the user API
- Example Matlab user application for reading all relative encoders on the robot arm
- It is not possible achieve high sampling loops when closing the loops in Matlab in non-real-time manner
- Proprietary software

Local PID Controller Template

- Highly experimental
- PID position loop template with set-point feed-forward
- Set-points and feed-forward gain can be send via USB using the Real-Time Data Communication Channel
- 2 position loops can be run at a sampling frequency close to 4 kHz
- No set point profiling! It Should be done on Linux/Windows/Matlab side
- **BE CAREFULL WITH MAXIMUM SPEED SPECS ESPECIALLY FOR SHOULDER**
- **Use at your own risk!!!**

Deliverables

- Robot Arm
- Desktop PC with Linux/Xenomai installation (incl. Keyboard & mouse)
- Linux Device Driver, User API and Example User Application
- Experimental Windows Driver and Matlab toolbox (“As Is”, use at own risk)
- Example Matlab user application(s) (“As Is”, use at own risk)
- 3rd party Windows application for uploading new firmware (inc. its documentation)
- Documentation
  - “Experimental Philips Robot Arm – User Manual”
  - “RT-Motion USB – User API Software Documentation”
  - Instruction day presentations

Questions?

PHILIPS