Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Embedded Debug Interface for Robots

Robin Mirow

23.06.2020

Supervisors: Dr. Andreas Mäder,
M. Sc. Marc Bestmann

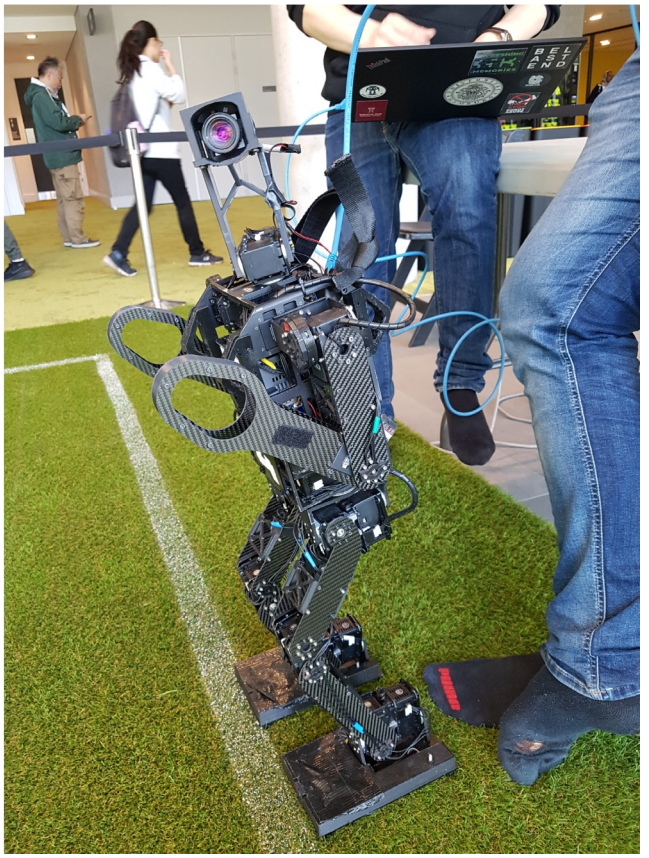
https://github.com/Laegluin/embedded_debug_interface_for_robots

Outline

1. Motivation
2. Related Work
3. ROBOTIS Dynamixel Protocol 2.0
4. Hardware
5. Implementation
6. Performance
7. Future Work

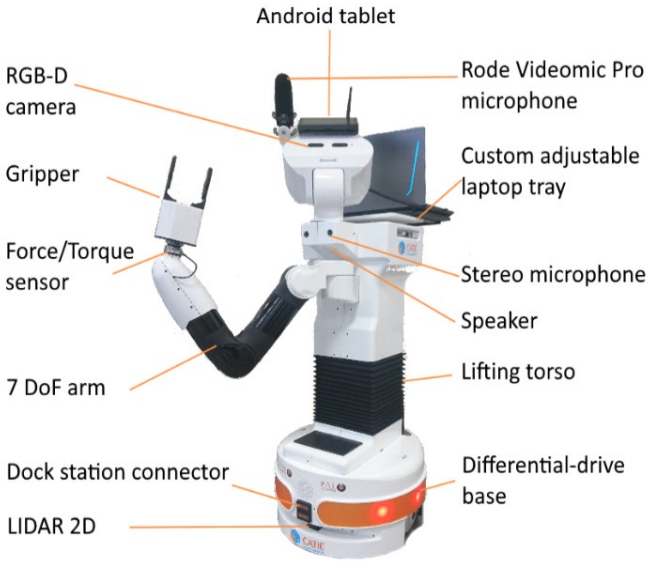
Motivation

- Wolfgang Robot Platform consists of many devices connected by a shared bus
- Computer must be connected via ethernet to diagnose issues
- An integrated display would make debugging in the field easier



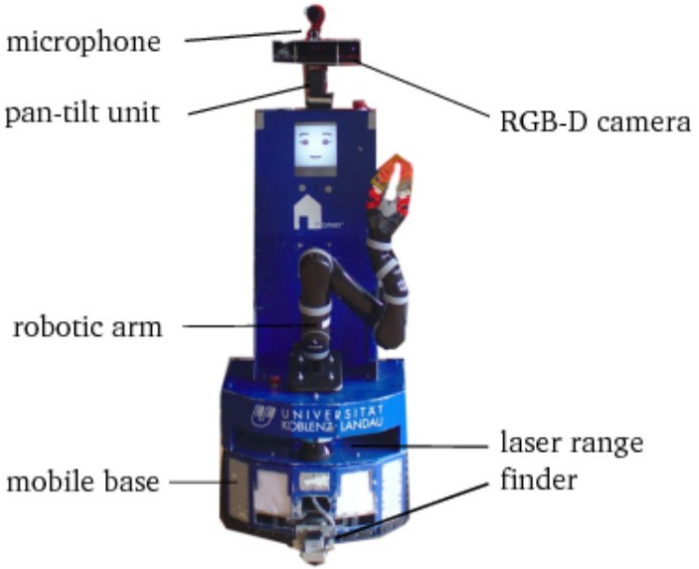
Related Work

RoboCup@Home Open Platform League 2019



CATIE Robotics

https://github.com/RoboCupAtHome/AtHomeCommunityWiki/wiki/files/tdp/2019-opl-catie_robotics.pdf



RT Lions

https://github.com/RoboCupAtHome/AtHomeCommunityWiki/wiki/files/tdp/2019-opl-homeratunikoblenz.pdf

homer@UniKoblenz

https://github.com/RoboCupAtHome/AtHomeCommunityWiki/wiki/files/tdp/2019-opl-rt_lions.pdf

Related Work

SoftBank Robotics *Pepper*

- Limited commercial success
- Focus on social interactions
- Highly customizable through software

https://www.softbankrobotics.com/emea/themes/custom/softbank/images/full-pepper.png

ROBOTIS Dynamixel Protocol 2.0

- Packet based master/slave protocol used for ROBOTIS products
- UART over RS-485 or TTL bus
- Built-in integrity check (CRC-16/BUYPASS)
- Each device is linear byte-addressed memory with 16-bit addresses
- Memory usually consists of memory-mapped registers

ROBOTIS Dynamixel Protocol 2.0

Packet layout

Name	Type	Content
Header	uint8_t	0xff
	uint8_t	0xff
	uint8_t	0xfd
Reserved	uint8_t	0x00
Device ID	uint8_t	Device ID
Length	uint16_t	Length
Instruction	uint8_t	Instruction
Payload	uint8_t	Byte 0
	...	...
	uint8_t	Byte n
CRC	uint16_t	CRC

Instruction packet layout

Name	Type	Content
Header	uint8_t	0xff
	uint8_t	0xff
	uint8_t	0xfd
Reserved	uint8_t	0x00
Device ID	uint8_t	Device ID
Length	uint16_t	Length
Instruction	uint8_t	Instruction
Error	uint8_t	Error
Payload	uint8_t	Byte 0
	...	...
	uint8_t	Byte n
CRC	uint16_t	CRC

Status packet layout

ROBOTIS Dynamixel Protocol 2.0

Example: *write* packet

Name	Type	Content
Address	uint16_t	Address
Data	uint8_t	Byte 0
	...	...
	uint8_t	Byte n

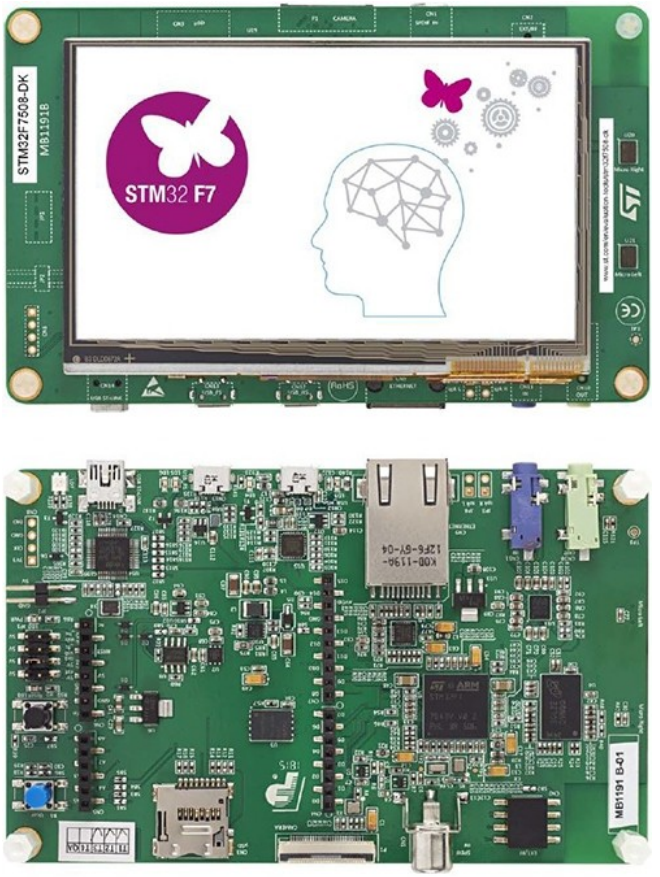
Instruction packet payload

Name	Type	Content

Status packet payload

Hardware

- STM32F7508-DK development board
- MAX485 RS-485/RS-422 transceiver
- FT232R USB to UART converter (for testing)



https://www.st.com/bin/ecommerce/api/image.PF267270.en.feature-description-include-personalized-no-cpn-large.jpg

Implementation

Libraries

- STM32F7 HAL and Low-layer drivers (part of STM32CubeF7 1.15.0)
- STM32F7508-Discovery board support package (part of STM32CubeF7 1.15.0)
- STemWin version 5.44 (part of STM32CubeF7 1.15.0)
- FreeRTOS version 10.0.1 (part of STM32CubeF7 1.15.0)
- Catch2 version 2.10.2

Implementation

Testing

- Most of the code only deals with arbitrary buffers
- Unit tests can be executed on the host platform
- IO must be tested manually
- FT232R USB to UART converter is used to simulate a bus by sending recorded data:
  - synthetic data
  - traces of real bus traffic

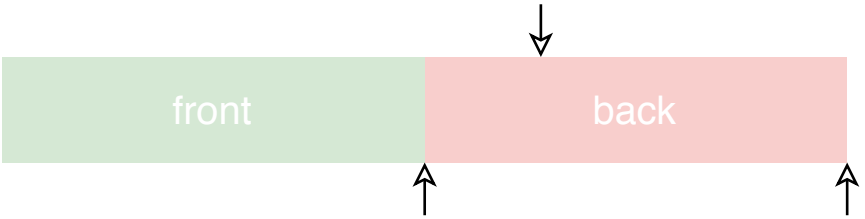
Implementation

Bootloader

- Board only has 64 KiB of internal flash
- Board also has 16 MiB of external flash
- Controller can only boot from internal flash
- An additional bootloader must program the external flash and start the application
- Bootloader can be controlled via USB and the board's *user* button

Implementation

Input buffering



- DMA controller transfers data to 8 KiB circular buffer
- Raises interrupts at half-full/full buffer
- Interrupt handler updates flag for currently valid half
- Software can process data while the other half is being written to

Implementation

Data processing

- Packets in incoming data are parsed
- Packets are analyzed and device IDs are mapped to control tables
- Responses to *ping* instructions determine the device model
- Devices that do not respond to instructions are considered disconnected

Implementation

Data processing

```
01 class ControlTable {
02   public:
03     /* some members omitted */
04
05     virtual uint16_t model_number() const = 0;
06
07     virtual const char* device_name() const = 0;
08
09     virtual ControlTableMemory& memory() = 0;
10
11     virtual const std::vector<ControlTableField>& fields() const = 0;
12
13     bool write(uint16_t start_addr, const uint8_t* buf, uint16_t len);
14
15     std::vector<std::pair<const char*, std::string>> fmt_fields() const;
16 };
```

Implementation

Data processing

```
01 const std::vector<ControlTableField> CoreBoardControlTable::FIELDS {
02     ControlTableField::new_uint16(
03         0,
04         "Model Number",
05         CoreBoardControlTable::MODEL_NUMBER,
06         fmt_number
07     ),
08     ControlTableField::new_uint8(2, "Firmware Version", 0, fmt_number),
09
10     ControlTableField::new_uint16(10, "LED", 0, fmt_bool_on_off),
11     ControlTableField::new_uint16(12, "Power", 0, fmt_number),
12     ControlTableField::new_uint32(14, "RGB LED 1", 0, fmt_core_rgb),
13     ControlTableField::new_uint32(18, "RGB LED 2", 0, fmt_core_rgb),
14     ControlTableField::new_uint32(22, "RGB LED 3", 0, fmt_core_rgb),
15     ControlTableField::new_uint16(26, "VBAT", 0, fmt_core_voltage),
16     ControlTableField::new_uint16(28, "VEXT", 0, fmt_core_voltage),
17     ControlTableField::new_uint16(30, "VCC", 0, fmt_core_voltage),
18     ControlTableField::new_uint16(32, "VDXL", 0, fmt_core_voltage),
19     ControlTableField::new_uint16(34, "Current", 0, fmt_core_current),
20     ControlTableField::new_uint16(36, "Power On", 0, fmt_core_power_on),
21 };
```

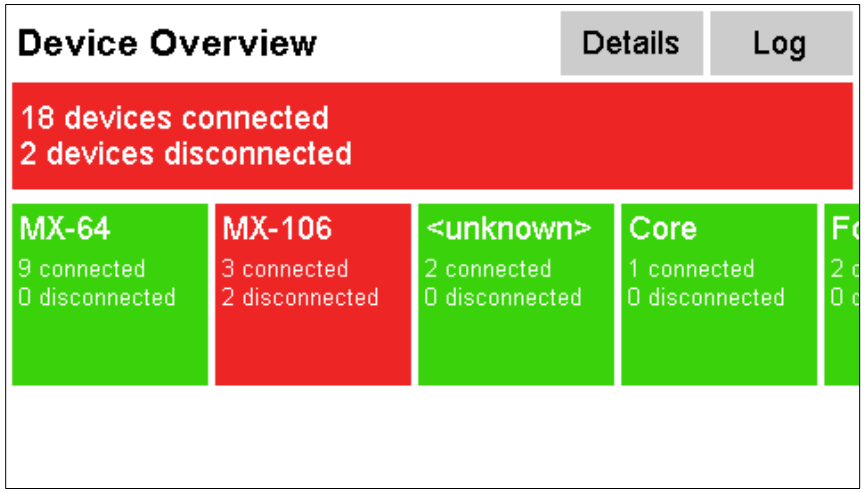
Implementation

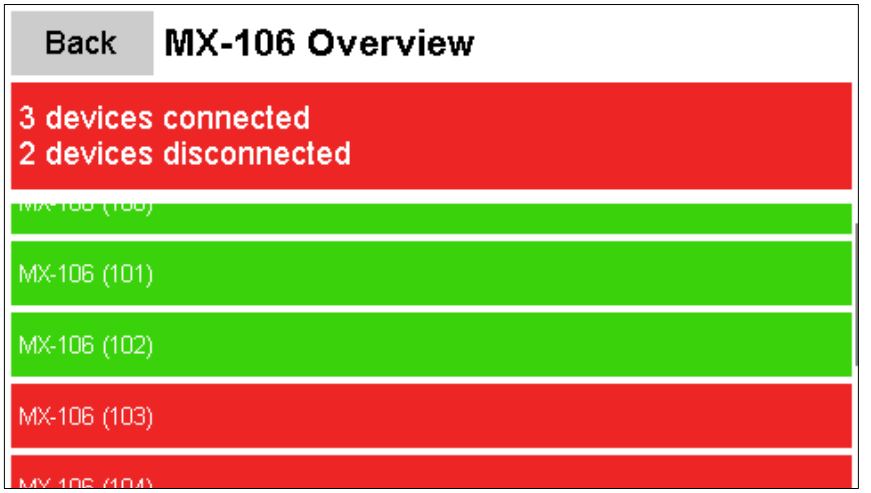
UI

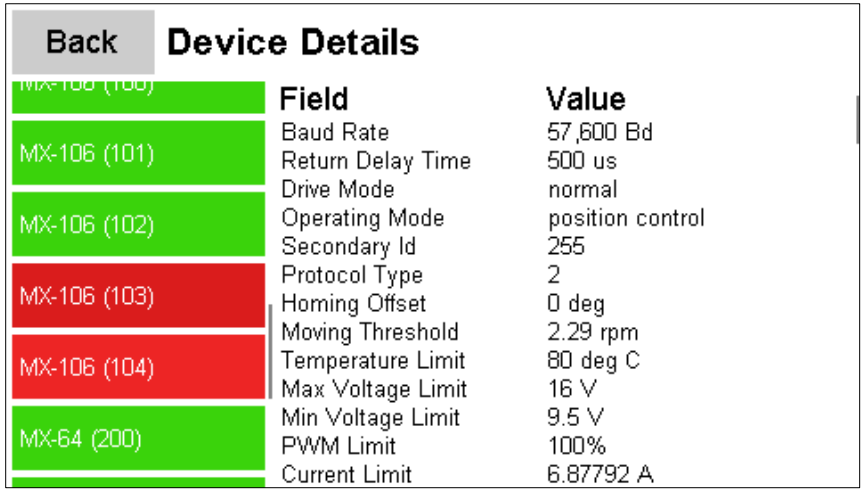
- Displays the values stored in each device's control table
- Makes it easy to identify disconnected devices at a glance
- Runs in a separate task
- UI updates should only lock state for a short amount of time to allow data processing to run

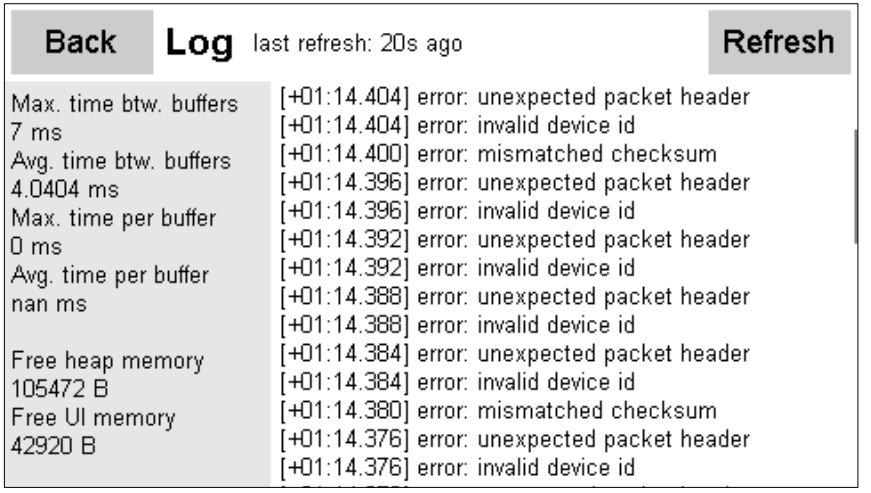
Implementation

UI









Performance

- Performance was measured by sending recorded data using the FT232R USB to UART converter
- Time per buffer
  - the time required to process one half of the receive buffer
- Time between buffers
  - the time between the starts of the processing for one half of the receive buffer

Performance

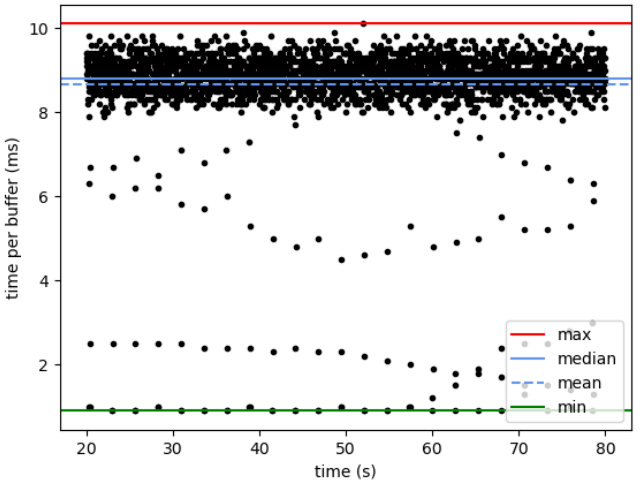
- Bus for the Wolfgang Robot Platform runs at 2 MBd

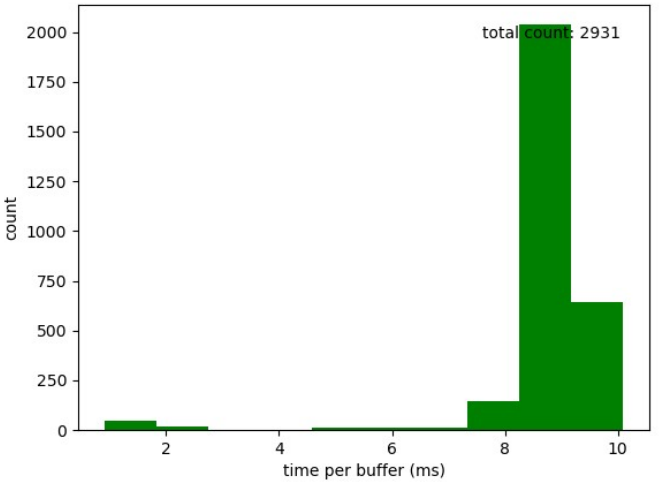
Maximum time between buffers without data loss:

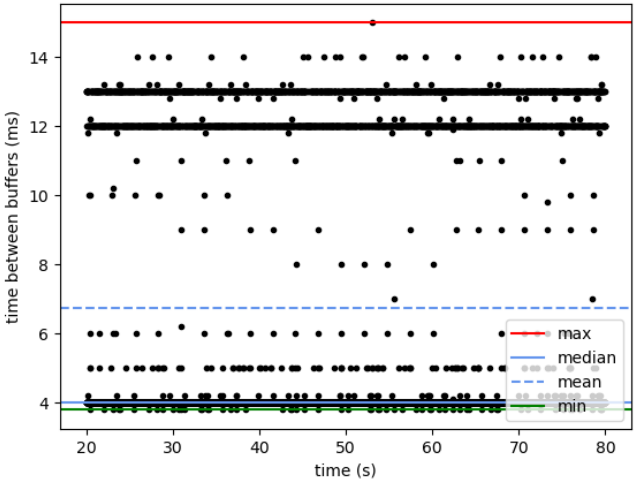
$$\frac{4096\,B}{2\,MBd} \approx 16\,ms$$

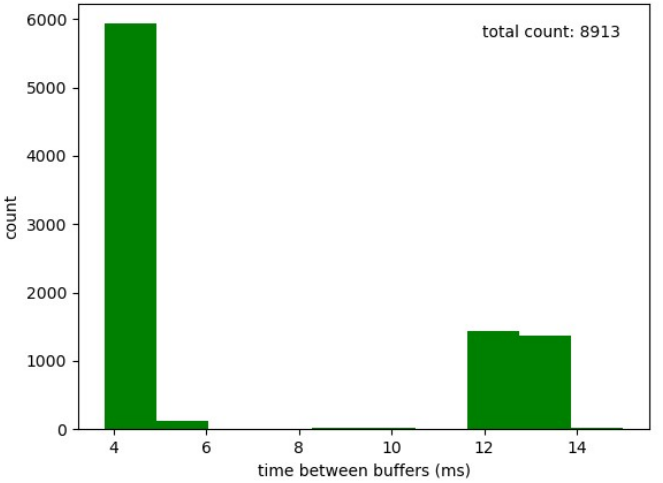
Performance

Trace



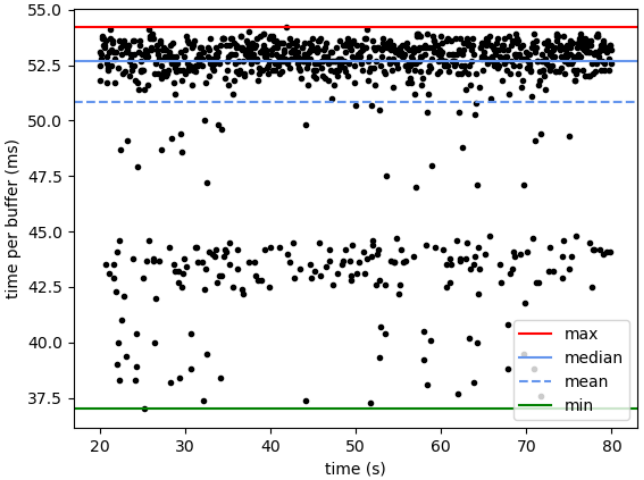


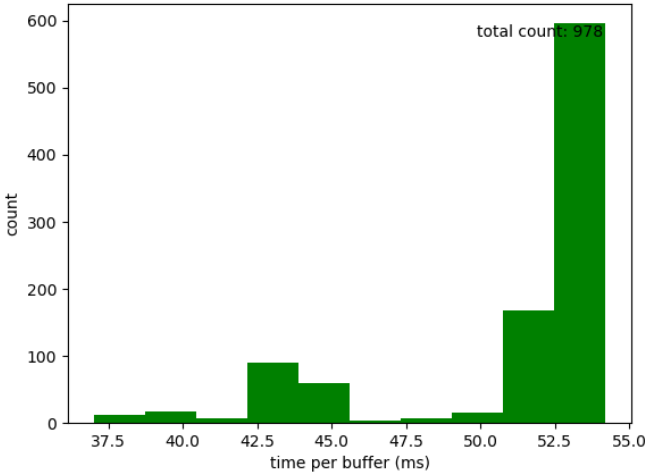


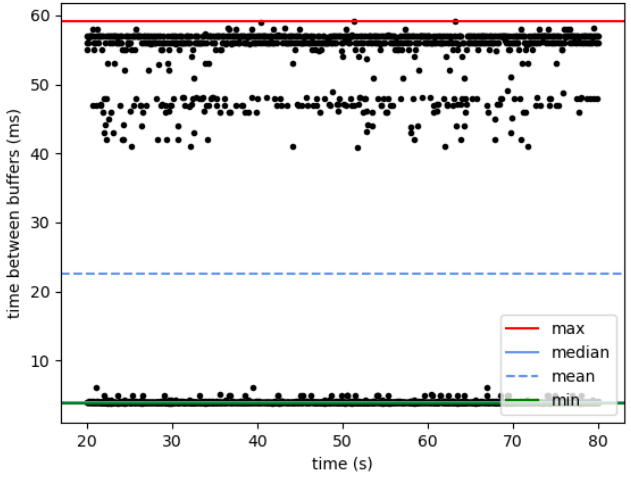


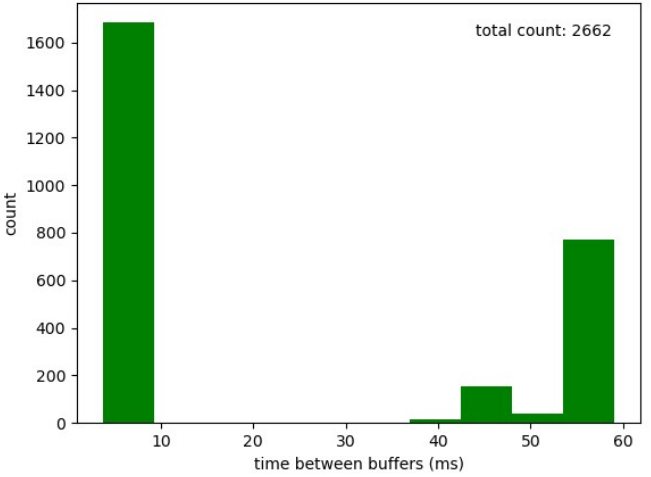
Performance

Synthetic *ping* instructions









Future Work

- Only show values that have been observed
- Accept data from more than one bus
- Increase performance
  - larger receive buffers
  - faster CPU
  - dual-core CPU
- Use V-Sync

Literature

ROBOTIS Dynamixel Protocol 2.0.
http://emanual.robotis.com/docs/en/dxl/protocol2/

Robots of the Hamburg Bit-Bots.
https://submission.robocuphumanoid.org/uploads/Hamburg_Bit_Bots-specs-5de3cecc85cbe.pdf

Qualified teams for RoboCup 2019.
https://humanoid.robocup.org/hl-2019/teams/

A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind.
A. K. Pandey and R. Gelin, IEEE Robotics Automation Magazine Vol. 25, pp. 40-48.

RS-422 and RS-485 Standards Overview and System Configurations.
Texas Instruments, Manny Soltero, Jing Zhang and Chris Cockril, June 2002.

Catalogue of parametrised CRC algorithms with 16 bits.
http://reveng.sourceforge.net/crc-catalogue/16.htm

Literature

RM0385 Reference manual - STM32F75xxx and STM32F74xxx advanced Arm®-based 32-bit MCUs.
STMicroelectronics, Revision 8, June 2018.

UM2470 User manual - Discovery kit for STM32F7 Series with STM32F750N8 MCU.
STMicroelectronics, Revision 1, October 2018.

DS12535 datasheet – STM32F750x8.
STMicroelectronics, Revision 1, June 2018.

MAX481/MAX483/MAX485/MAX487–MAX491/MAX1487 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers.
Maxim Integrated, Revision 10, September 2014.

Future Technology Devices International Ltd. FT232R USB UART IC Datasheet.
Future Technology Devices International, Version 2.15.

Literature

UM1734 User manual - STM32Cube™ USB device library.
STMicroelectronics, Revision 4, February 2019.

UM1905 User Manual - Description of STM32F7 HAL and Low-layer drivers.
STMicroelectronics, Revision 3, February 2017.

UM1891 User manual - Getting started with STM32CubeF7 MCU Package for STM32F7 Series.
STMicroelectronics, Revision 9, February 2019.

Release Notes for STemWin Library.
STMicroelectronics, March 2018 (Part of STM32CubeF7 1.15.0).

The FreeRTOS™ Kernel.
https://www.freertos.org/RTOS.html

Why do we need yet another C++ test framework?
https://github.com/catchorg/Catch2/blob/87950d9cfa87eb41ff60b7e5f7e11ad21749a2a1/docs/why-catch.md