Weekly Presentation

Week 40

Luleå University of Technology

September 29, 2020

Group members

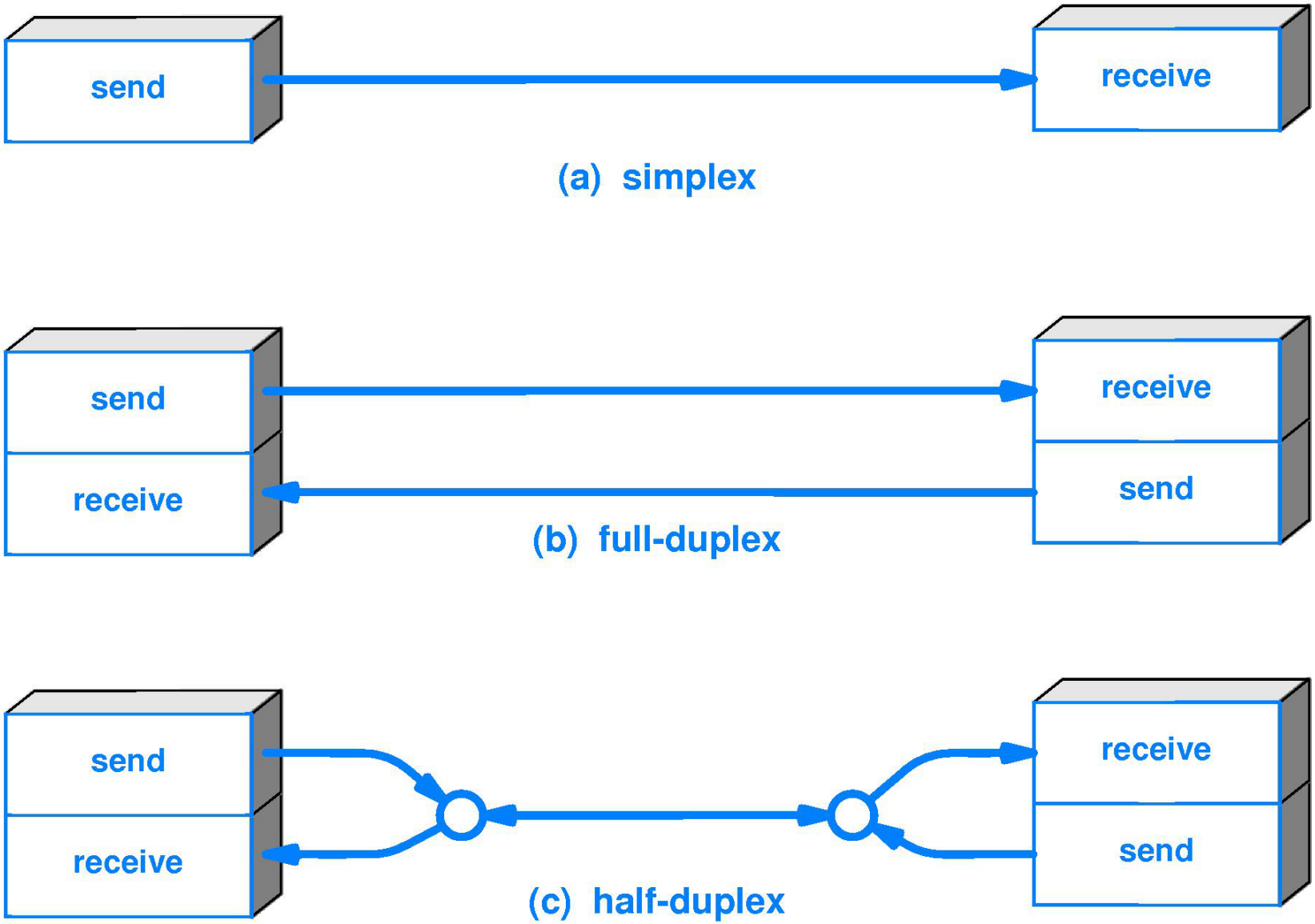
- Y-students
  - Martin Blaszczyk - Project leader and object detection
  - Edward Cedergård -Arm and gripping tool
  - Niklad Dahlqvist - Arm and gripping tool
  - Måns Norell - Movable base
- D-students
  - Edward Källstedt - Object detection
  - Albin Martinsson - Arrowhead and Git

Overview

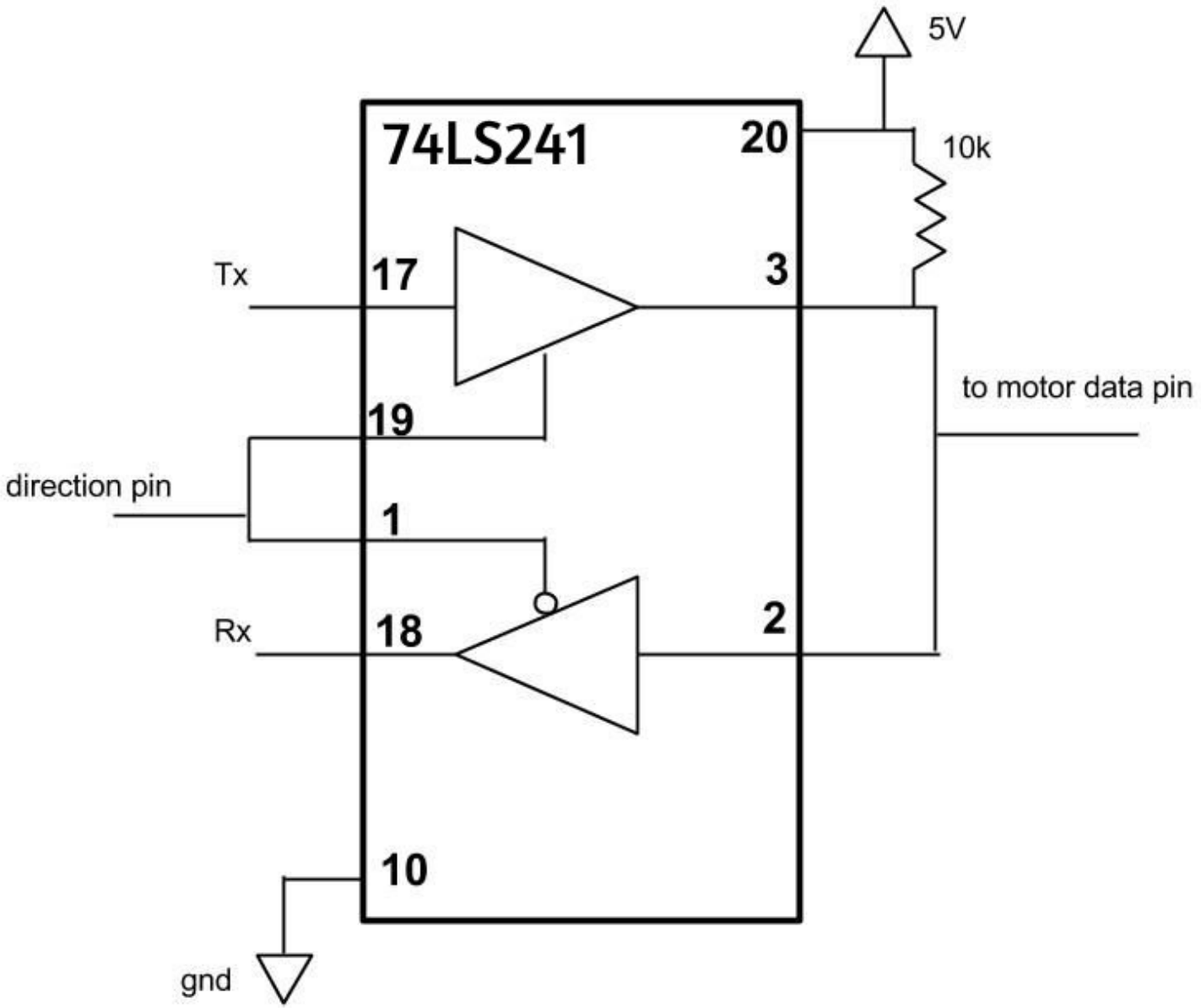
What we have done and what we are working on:

- Full duplex to half duplex serial communication
- Dynamixel data packages

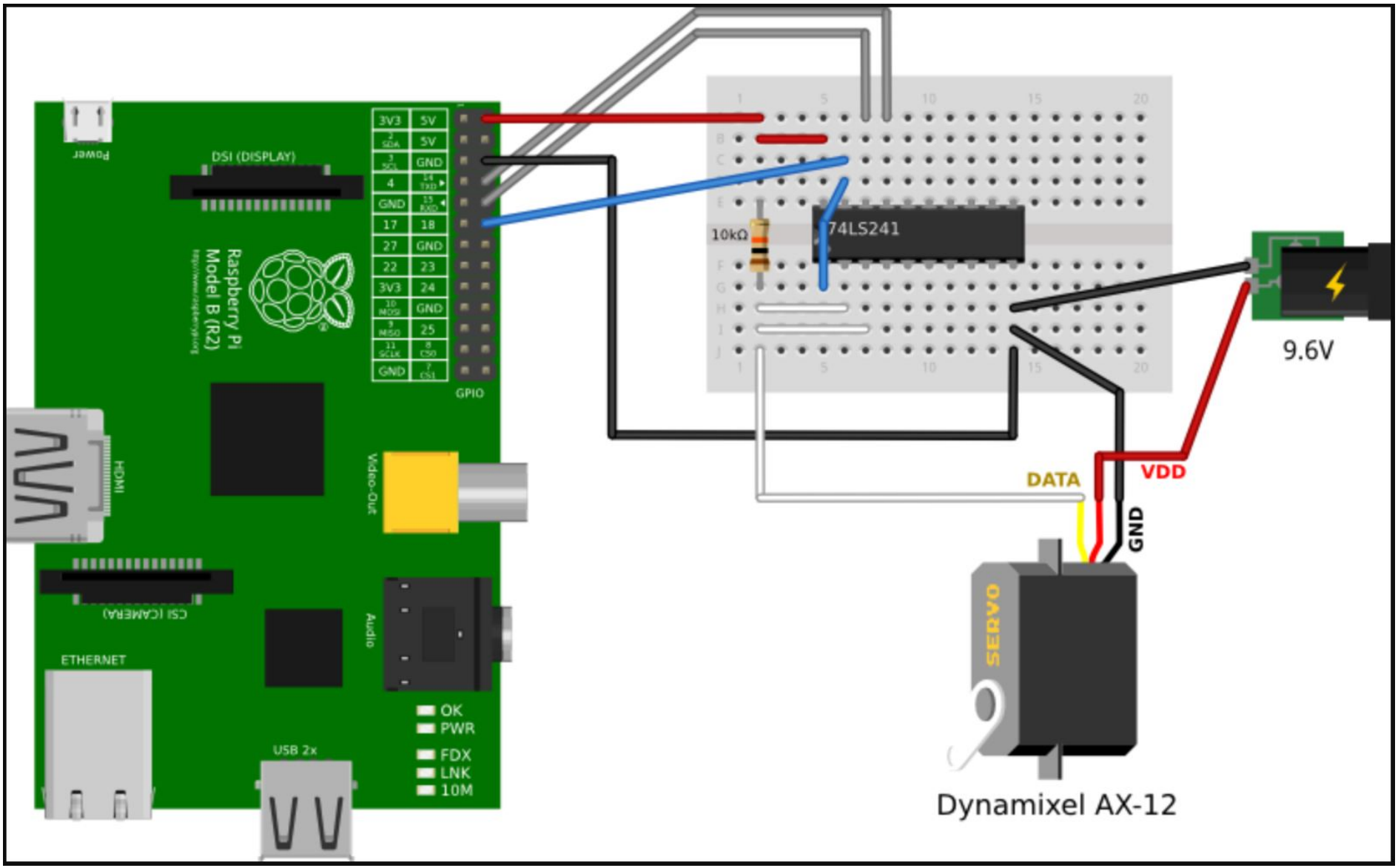
Full-duplex to half-duplex



Dynamixel communication



Curcuit


DSI (DISPLAY)
Raspberry Pi
Model B (R2)
GPIO
HDMI
Video-Out
CSI (CAMERA)
Audio
ETHERNET
USB 2x
OK
PWR
FDX
LNK
10M
10kΩ
74LS241
9.6V
DATA
VDD
GND
SERVO
Dynamixel AX-12

Dynamixel data packages overview

- Data packets structure
- Timing of response
- Example package

Dynamixel data packages

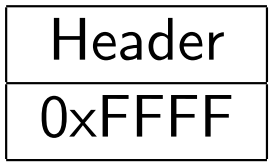
Instruction package - send to the motor

Header	ID	Length	Instruction	Param 1	...	n	Checksum
0xFFFF	ID	Length	Instruction	param 1	...	n	Checksum

Status return package - recieve from the motor

Header	ID	Length	Error	Param 1	...	n	Checksum
0xFFFF	ID	Length	Error	Param 1	...	n	Checksum

Instruction package



Header allways fixed.

Instruction package

Header	ID
0xFFFF	ID

ID is a unique number for each motor connected.

Instruction package

Header	ID	Length
0xFFFF	ID	Length

Length of the message, excluding the header bytes.

Instruction package

Header	ID	Length	Instruction	Param 1	...	Param n
0xFFFF	ID	Length	Instruction	param 1	...	Param n

Parameters, depends on the instruction.

Instruction package

Header	ID	Length	Instruction	Param 1	...	n	Checksum
0xFFFF	ID	Length	Instruction	param 1	...	n	Checksum

The checksum is calculated as
Checksum = (ID + Length + Instruction + Parameter1 + . . .
Parameter N)
where "" is the "not" operation and only the lower byte is used.

Status return package

Header	ID	Length	Error	Param 1	...	Param n	Checksum
0xFFFF	ID	Length	Error	Param 1	...	Param n	Checksum

Similar to instruction package but each bit in the error byte represents one possible error.

Timing of return package

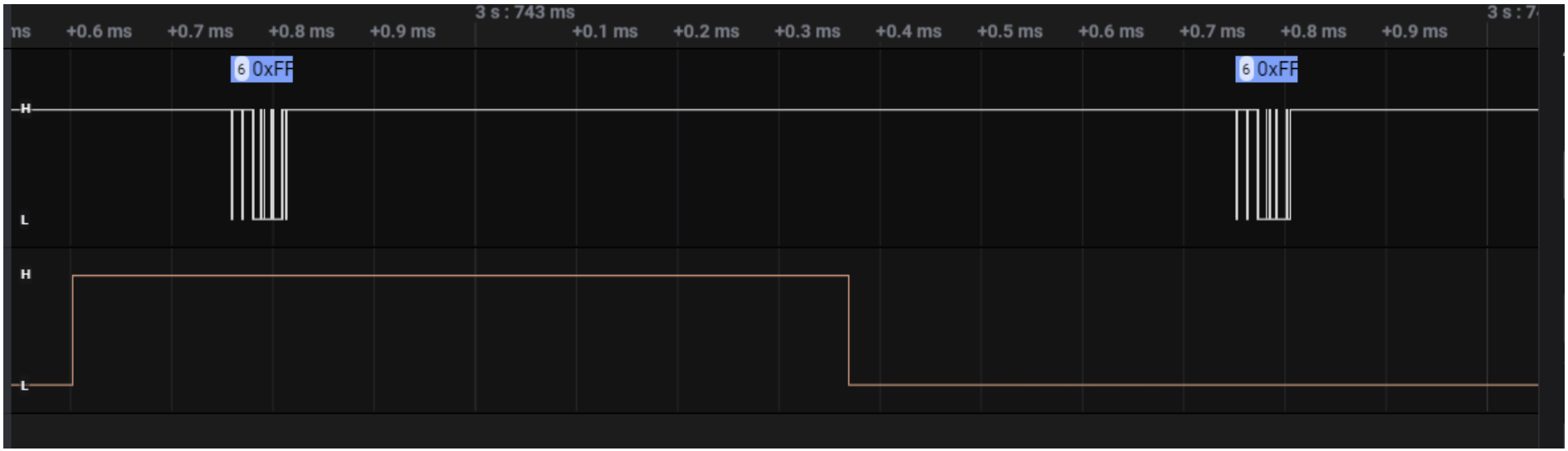
- Return delay can be set for each motor
- Values between 0 - 508 microseconds

Example package

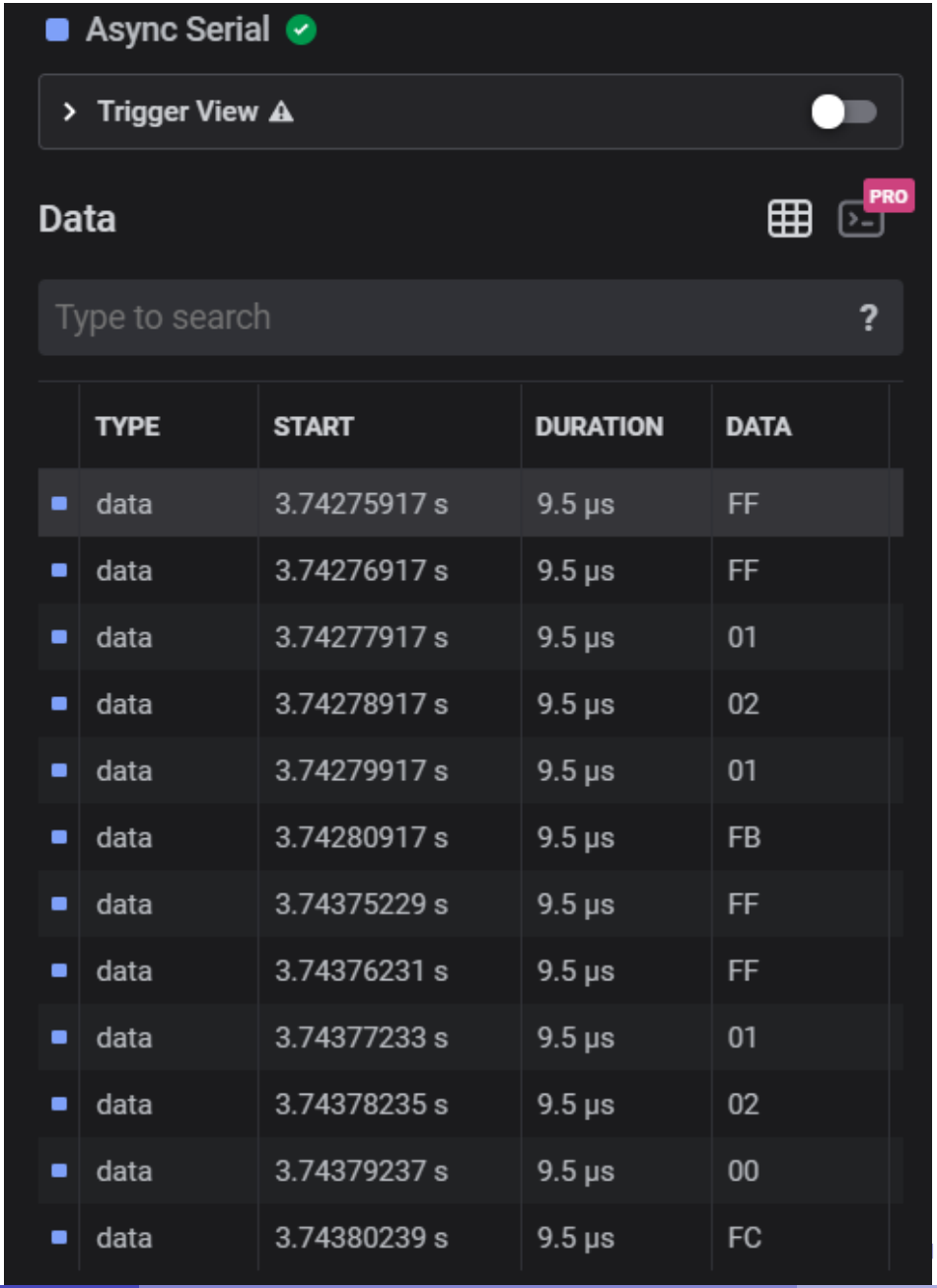
Header	ID	Length	Instr.	P.1	P.2	P.3	Checksum
0xFFFF	0x01	0x04	0x03	0x1E	0x00	0x00	0xD9

Writes the data 0x0000 at memory position 0x1E. (Goal position = 0)

Logic analyzer example



Logic analyzer example



TYPE	START	DURATION	DATA
data	3.74275917 s	9.5 µs	FF
data	3.74276917 s	9.5 µs	FF
data	3.74277917 s	9.5 µs	01
data	3.74278917 s	9.5 µs	02
data	3.74279917 s	9.5 µs	01
data	3.74280917 s	9.5 µs	FB
data	3.74375229 s	9.5 µs	FF
data	3.74376231 s	9.5 µs	FF
data	3.74377233 s	9.5 µs	01
data	3.74378235 s	9.5 µs	02
data	3.74379237 s	9.5 µs	00
data	3.74380239 s	9.5 µs	FC

Demonstration

- https://youtu.be/f_JAT8srcIc

Additional information

- https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/
- https://emanual.robotis.com/docs/en/dxl/protocol1/
- http://www.crustcrawler.com/products/bioloid/docs/AX-12.pdf

Movable Base

What has been done

- CAD model for base have been made
- Controllers have been made to move the base to a specific point in space
- Real world limitations have been applied to the controllers

CAD Model

THIS IS NOT THE FINAL BASE MODEL
It's for prototyping and is modelled to be modular and easy to attach parts to

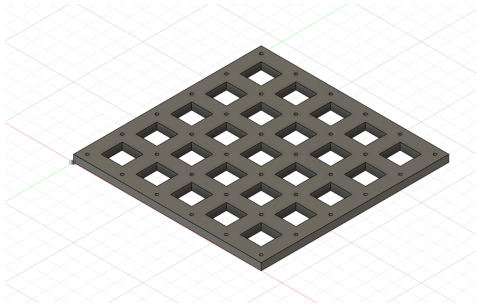


Figure: Model of base

New Controllers

Two **PID** controller are made in order to move to a user specific coordinate by

- Setting the robot to a specific angle
- Move the robot a user defined distanced

Simulations With Limitations

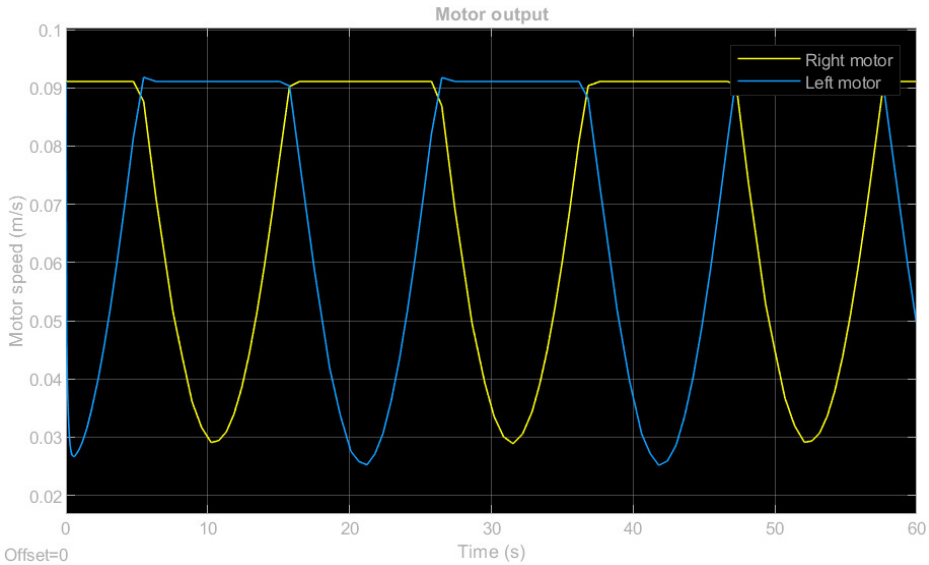


Figure: Motor speeds

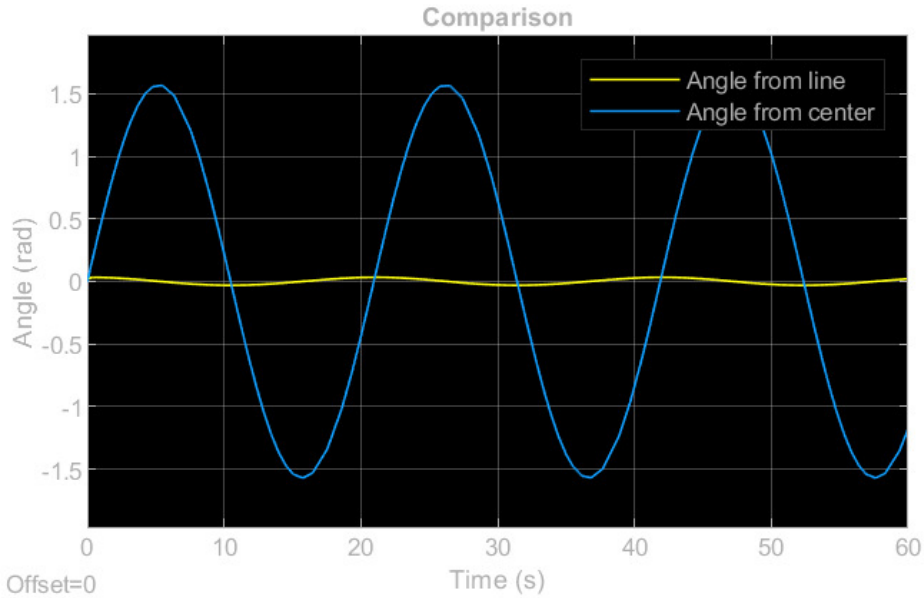


Figure: The angle towards the center and the angle towards the line comparison

Simulations With Limitations

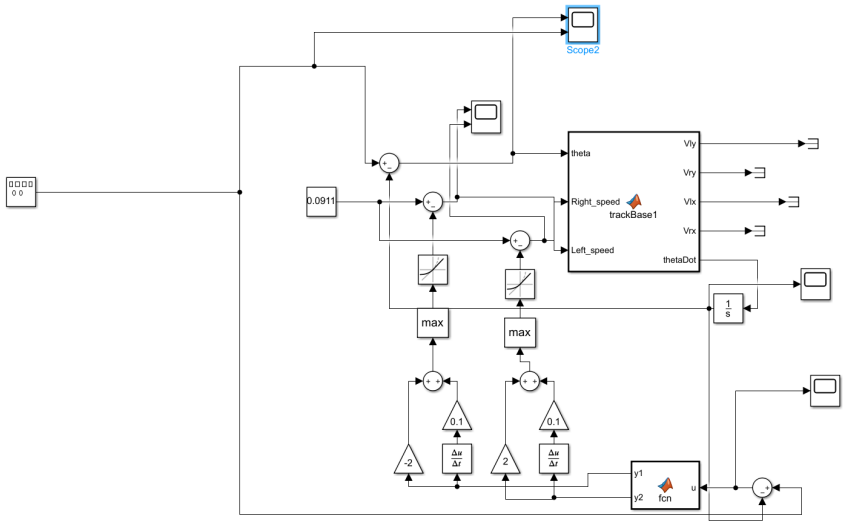


Figure: Block diagram

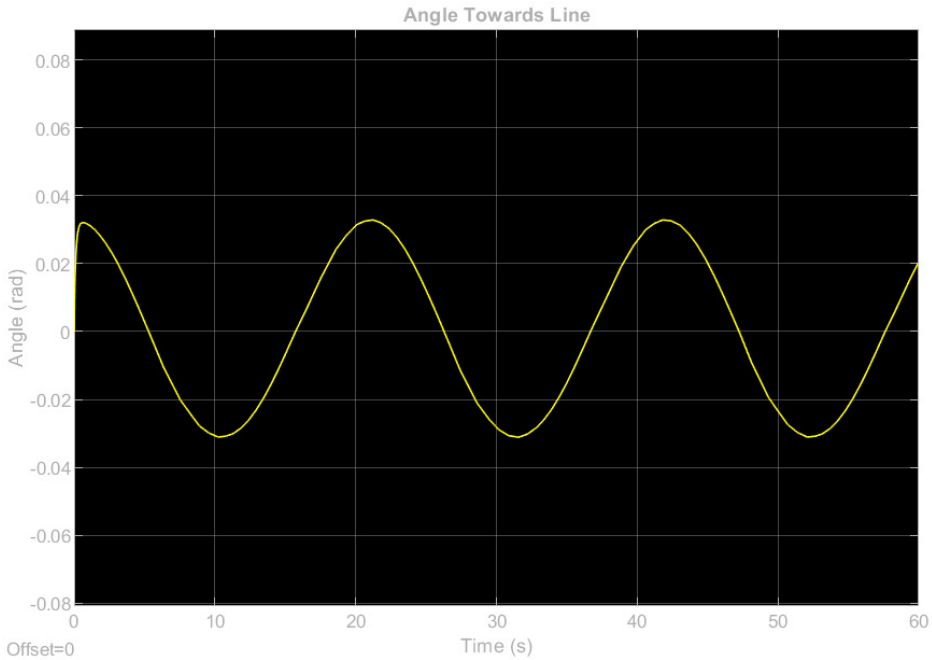


Figure: The angle towards the line

What Is To Be Done

- Stability analysis on **PID**s
- 3D print and build robot
- Implement controllers on robot

Machine Vision Progress

- Rewritten in C++
- Added configuration options
- QR-Code detection implemented

Machine Vision Progress

- https://www.youtube.com/watch?v=b7bSlFM6s4o

Machine Vision Future Improvements

- QR detection optimizations
- QR orientation
- Pathfinding

Overall timetable

Sep	Oct	Nov	Dec
Concept generation	Evaluation	Evaluation	
Theory	Prototyping	Evaluation	Finishing up
Simulation	Evaluation	Evaluation	
Prototyping	Final Design	Evaluation	

Time plan for September

Subproject	Week 1	Week 2	Week 3	Week 4
Arrowhead	Reading	Setup	API	Prototyping
Movable base	Reading	Modeling	Simulation	Implementation
Arm and grip	Reading	Kinematics	Simulation	Prototyping
Object detection	Reading	Testing	Prototyping	Evaluation

Questions?