Practical Use of FPGA Chips for Implementation of Linear Motor Control System

Design and Implementation Control System that Benefit from Advantages of FPGA Chips

Ing. Matěj Bartík

Department of Digital System Design
Faculty of Information Technology
Czech Technical University in Prague

12. 6. 2014

Problem introduction — Initial setup

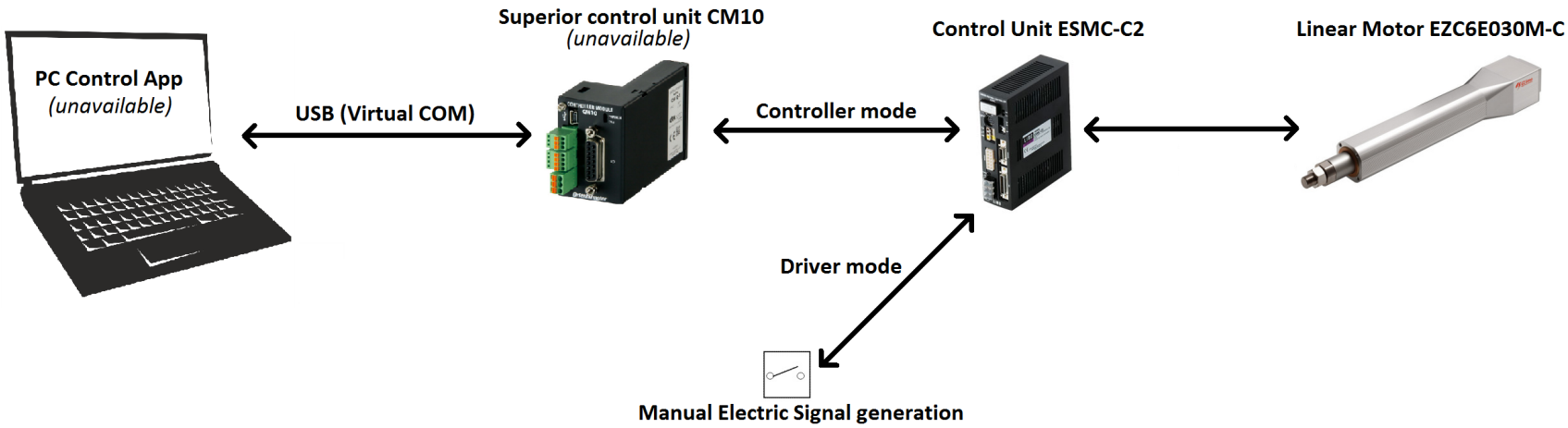


Figure : Initial setup of linear motor and all its parts

Requirements of control system funcionality

- Ability to perform complex movements with dynamic high speed and high range of distance.
- Movements should be smooth for simulating general curves.
- Ability to perform long–time tests (more than a week).
- Ability for reliable and precise measurements.
- Software part should be implemented in MATLAB enviroment.
- Control system support for colaboration with other equipment in laboratory.
- Control system should react to unexpected events with low latency.
- Contol system should be universal for same vendor motors.

Approach no. 1 — Reverse engineering

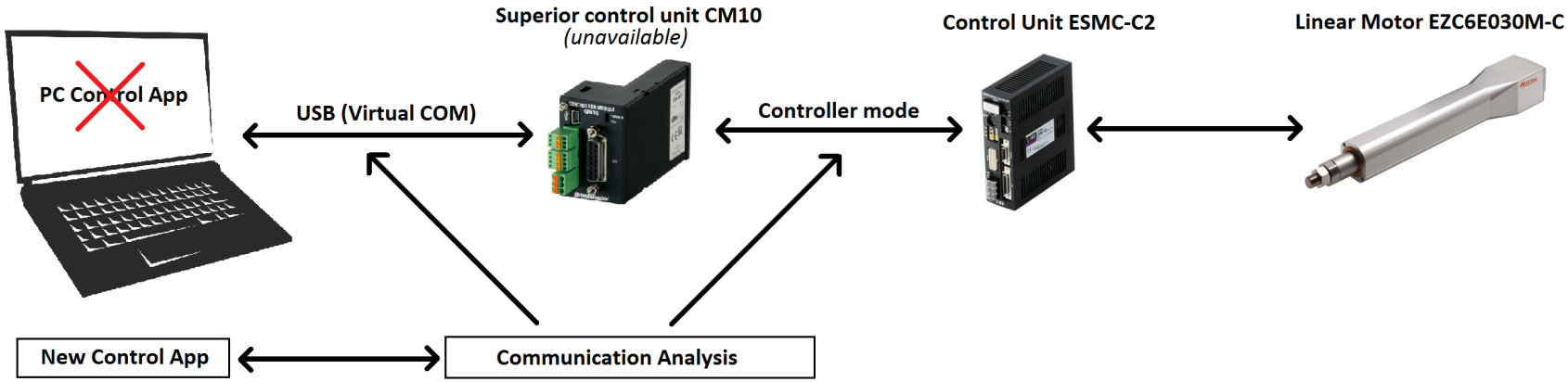


Figure : Expected state after reverse engineering

Approach no. 2 — Completely new system

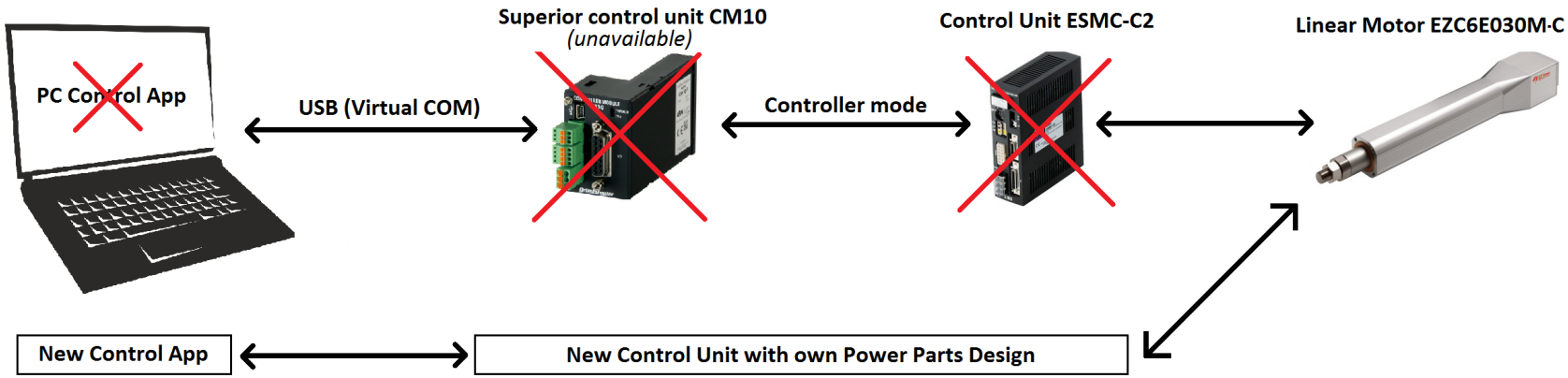


Figure : Expected state after developing of completely new system

Approach no. 3 — Reusing all suitable parts of current setup

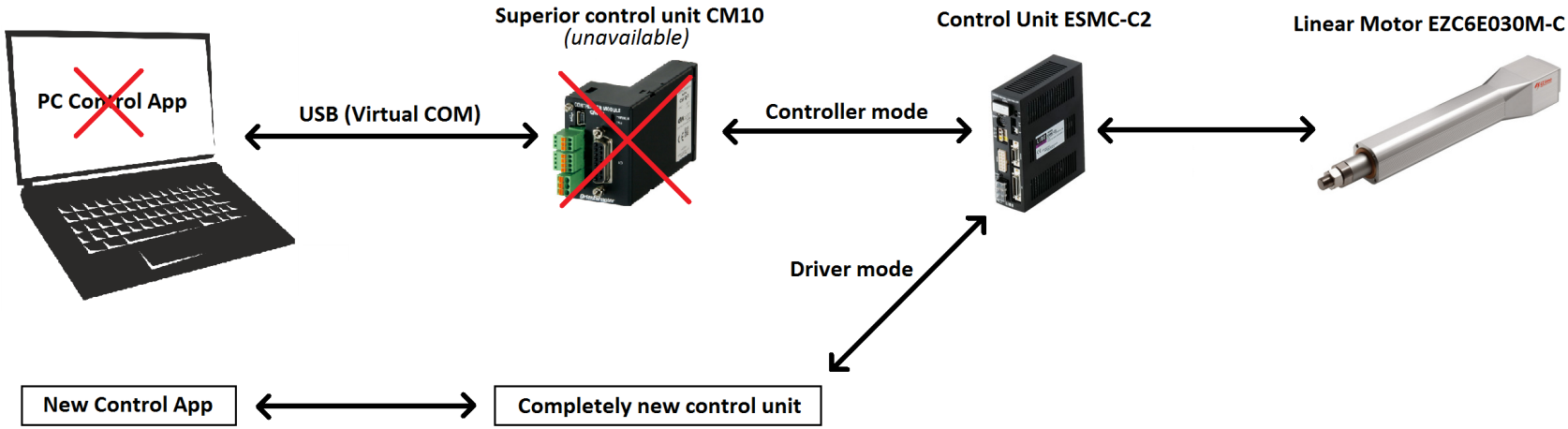


Figure : Expected state after reusing all suitable parts of current setup

Layers of Control System

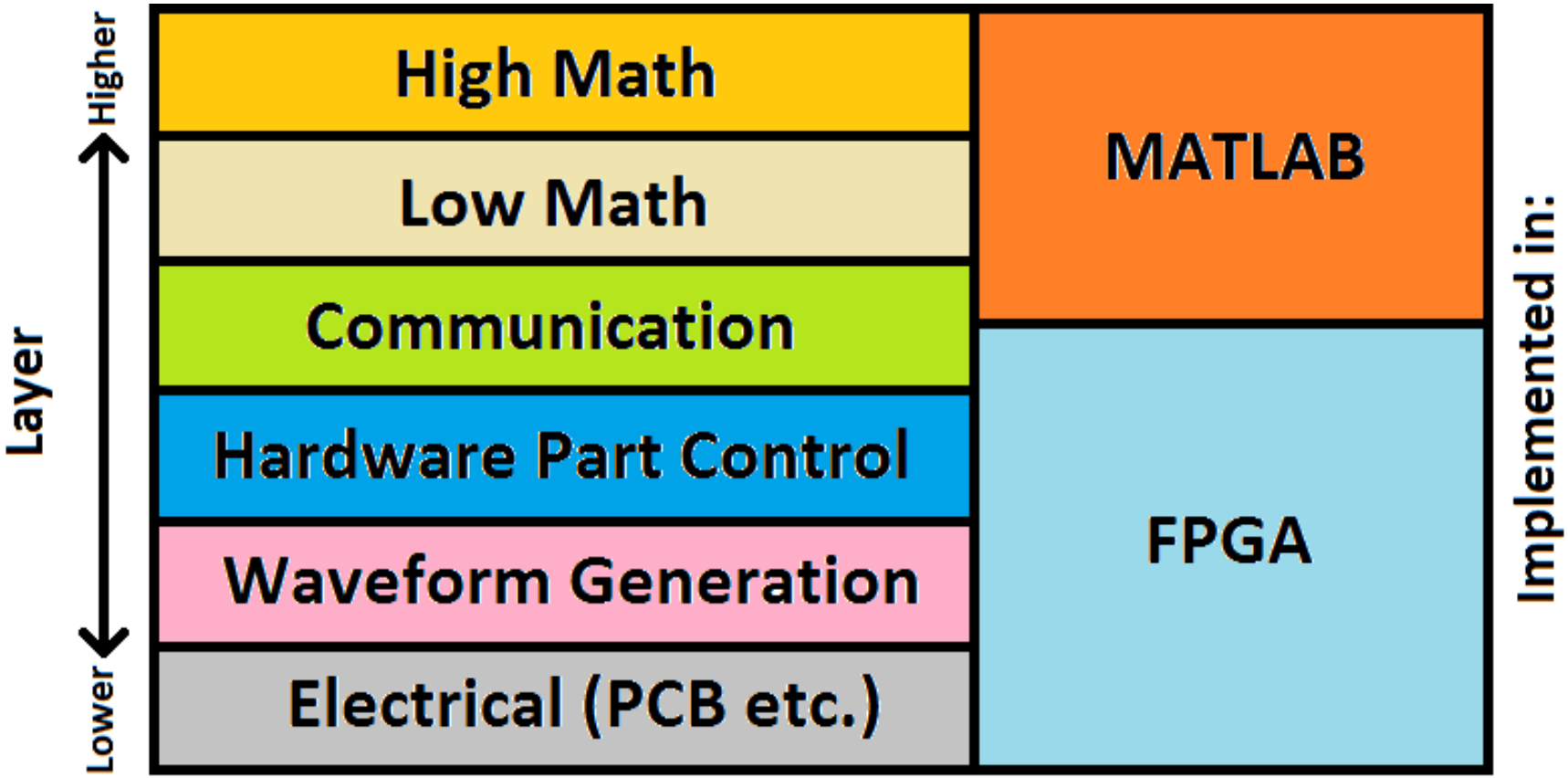


Figure : Layers planned for implementation in current design

Choice of Development Platform

Platform	Pros	Cons
	Price, easy to develop	Slow, not enought I/O pins, complex Program.
	Fast enought	Slow peripherals. Harder for development, inability to determine exact timing and behavior
	High speed, parallel executing of Program, easy to predict and guarantee timing and behavior	Hardest for development and high price.

Evaluation of sequential system execution Model

Problems of sequential system execution Model:

- Many complex steps, that take much time and have various lenght. They can't be predicted thanks to effect of Caches and Compiler.
- Shared memory for all data required deny to parallelize instruction execution and other stuff.
- Very complex errors and events handling, both are sequential.
- Many events, that can be missed (can be partially solved by RTOS).

In FPGA any of these things shouldn't have to be a problem!

Other advantages of FPGA Architecture

- All major parts are synchronous and can work independently.
- BlockRAMs can be divided across the whole architecture as local memories.
- All major parts are serialized (systolic algorithm).
- Internal events can be handled in a single cycle.
- External events are handled by dedicated hardware.
- Large queue allow to store huge amount of instructions.

Using queues for instructions

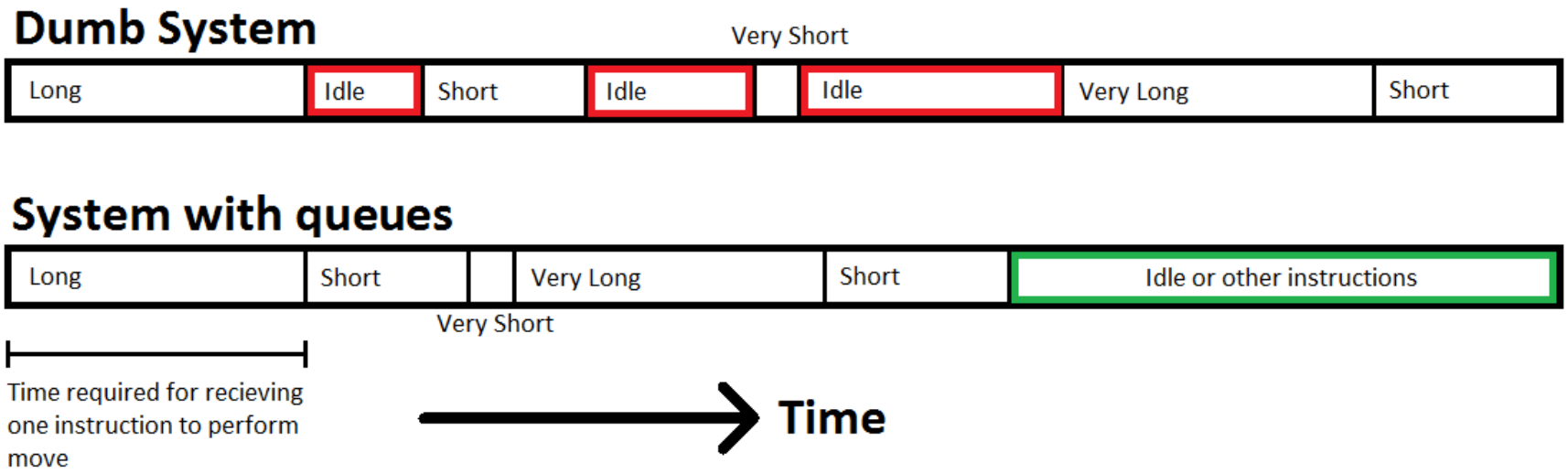


Figure : Queue making movements smooth

Simplified System Flow

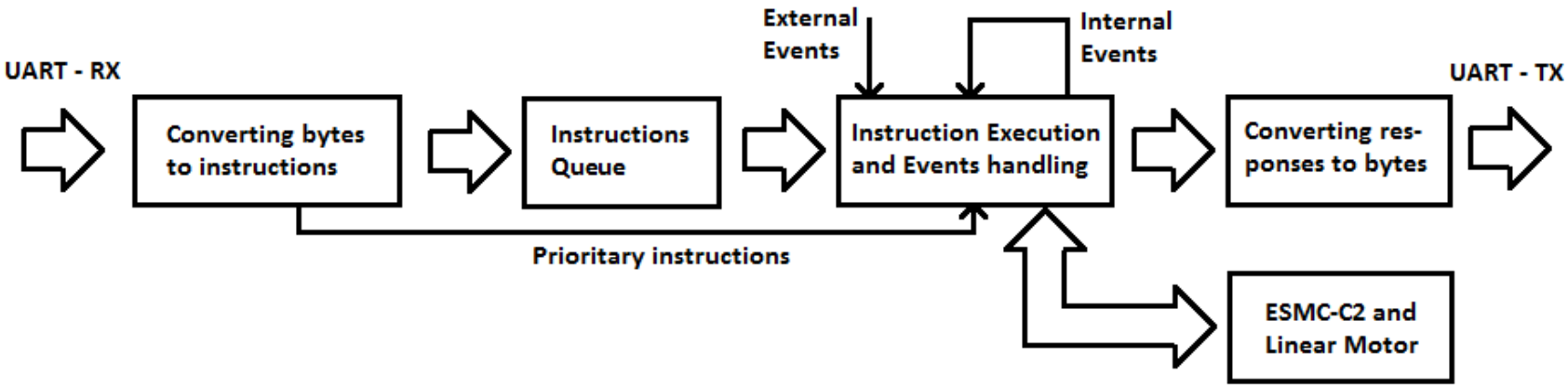


Figure : Simplified System Flow

Block Diagram of Execution Module

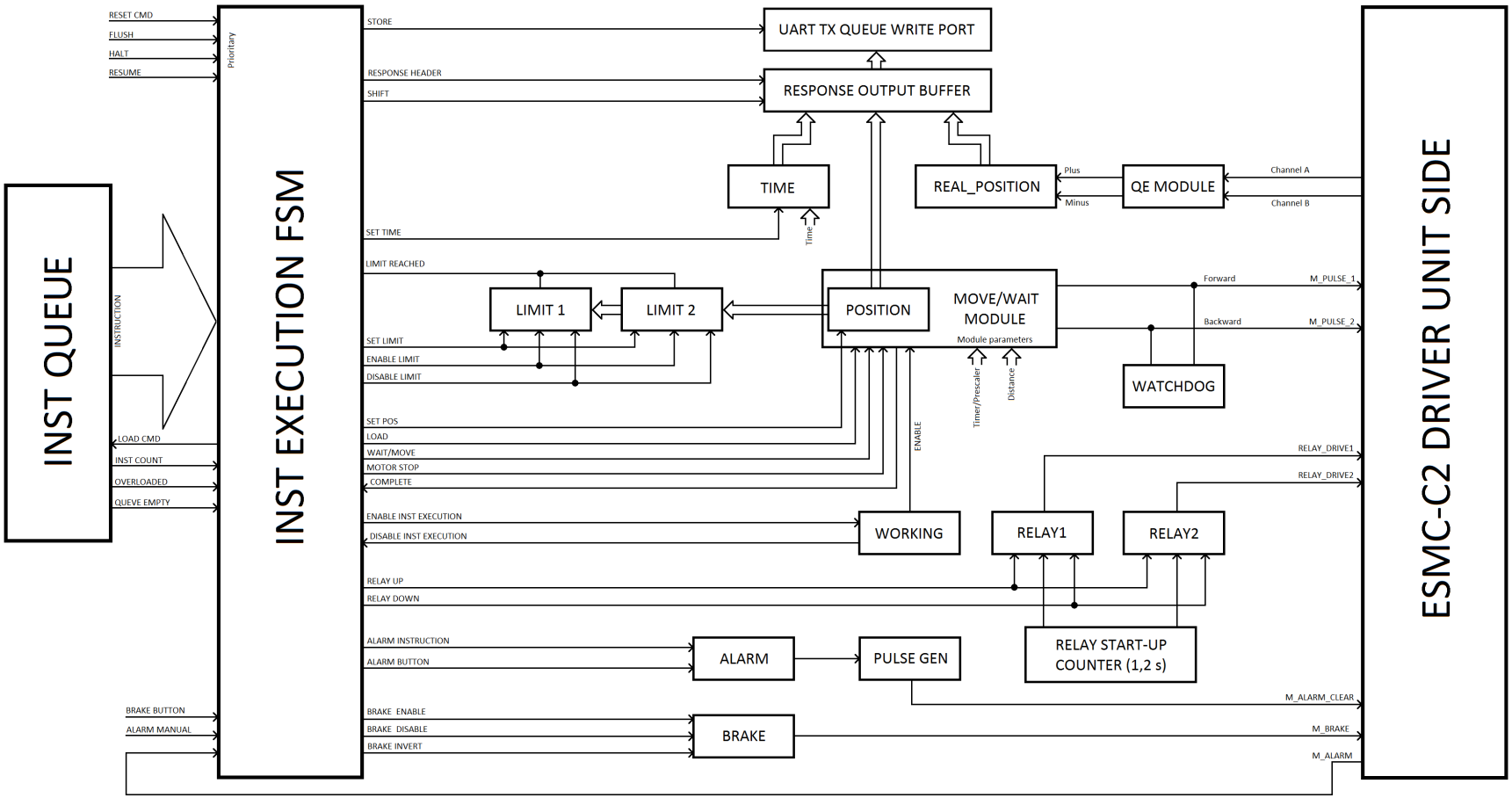


Figure : Block Diagram of Execution Module

Simplified Diagram of Instruction and Events Execution

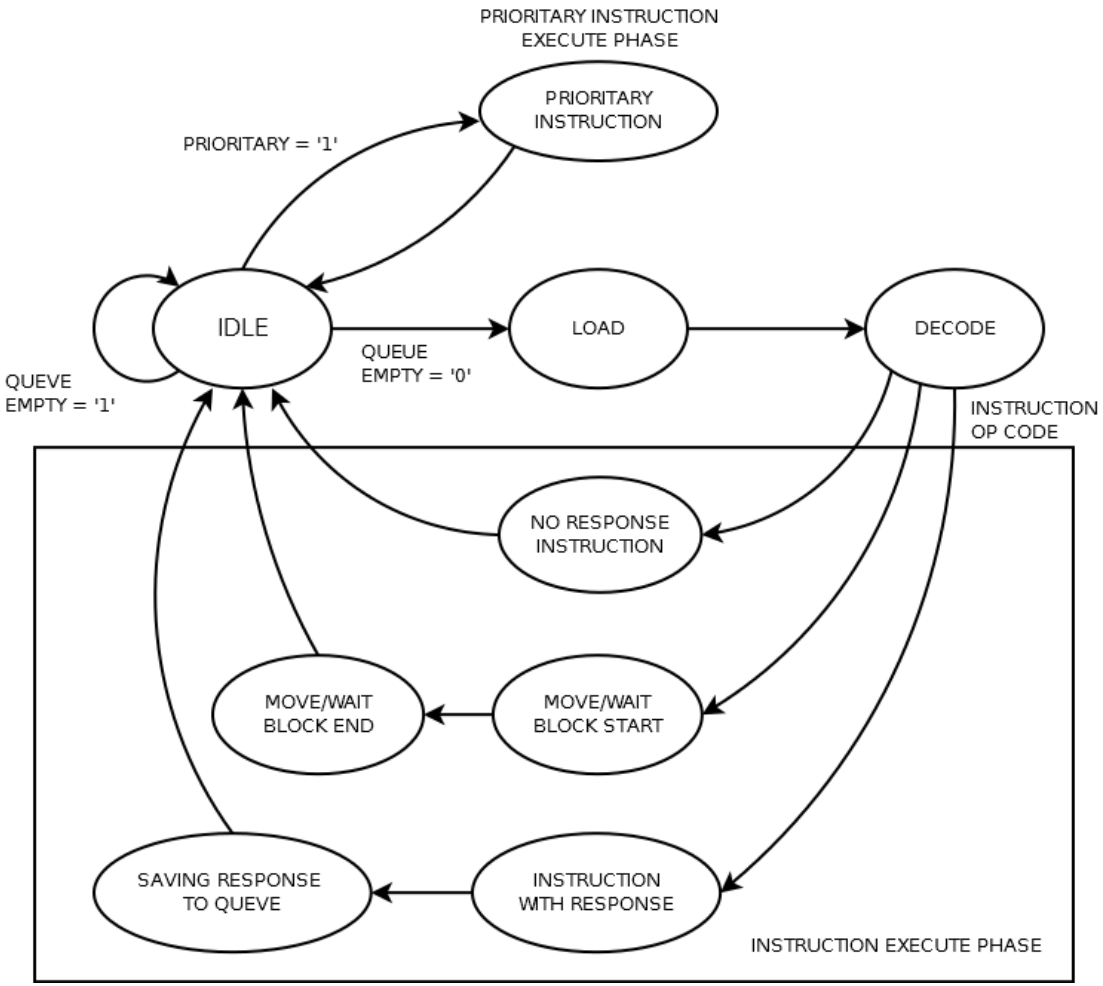
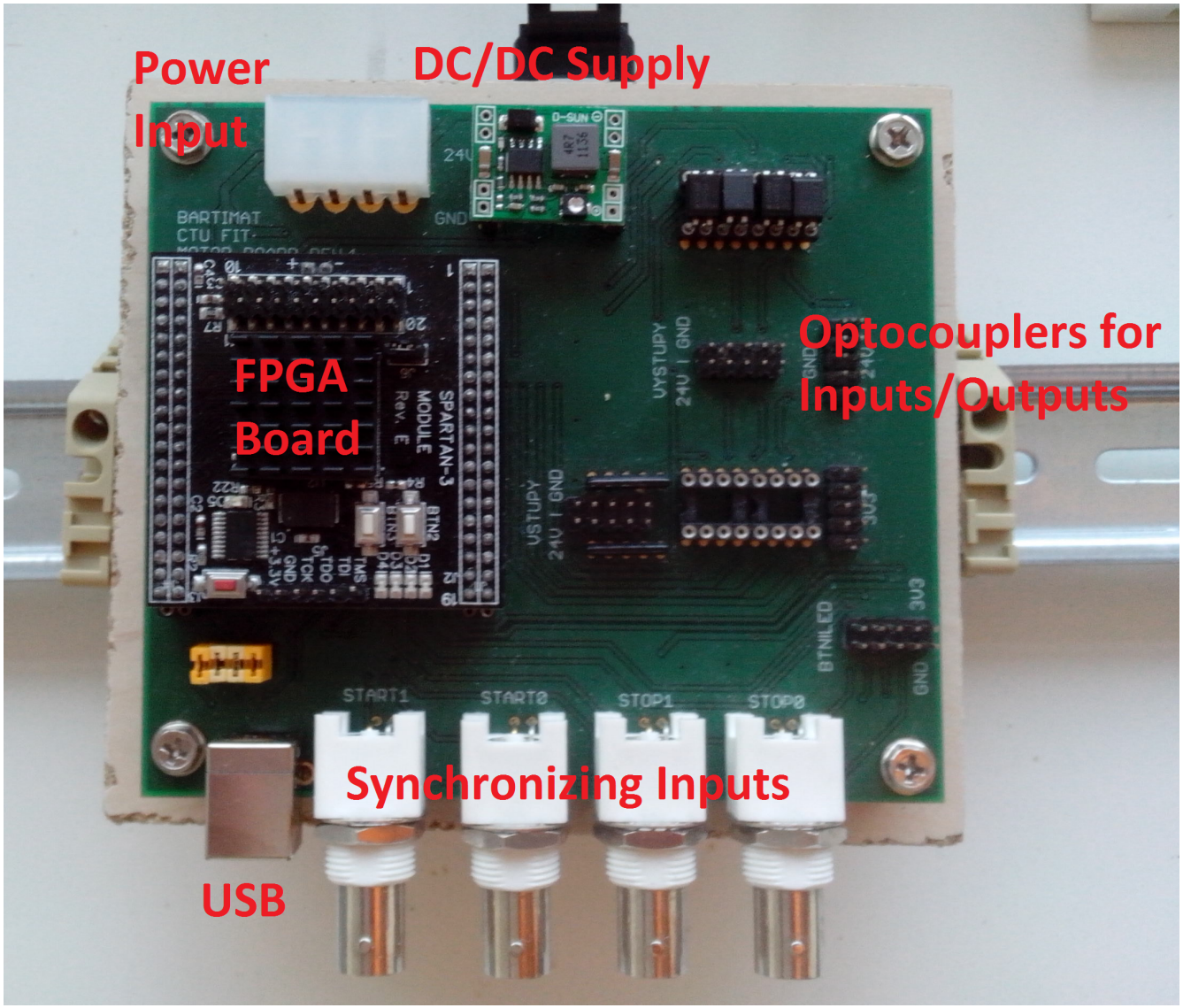


Figure : Simplified FSM for executing instructions

Implementation Summary

- Realized FSM have 48 states, 108 transition edges, 21 inputs and 37 outputs.
- This FSM can handle 27 instructions (divided into 4 classes) with 8 priority levels, generating 22 types of response.
- Every error or event generates response. Every error handling are fail-safe — It stops the Motor.
- Implemented functionality of "general stop", that can be activated in less than 100 μs. Its important for keeping tested samples in one piece during unpredicted events.

Implementation Summary — Realized Control System



Testing

- Every part of system were tested by performing long–time test runs.
- System was quite unstable – that can be caused by several different reasons.
  - Bug in code (SW or HW)
  - Bad synthesis — VHDL can be translated to HW, thats not performing expected funcionality.
  - Predisposition to Electromagnetic Interference.
- After 2 months of testing SW and HW part of Control System I focused on Electromagnetic Interference.
- By next 6 months of testing I was able to state that system unstabilty was caused by failure to comply with design rules of Printed Circuit Board (PCB doesn't met the EMC directives).

Testing Control App — Main Funcionality

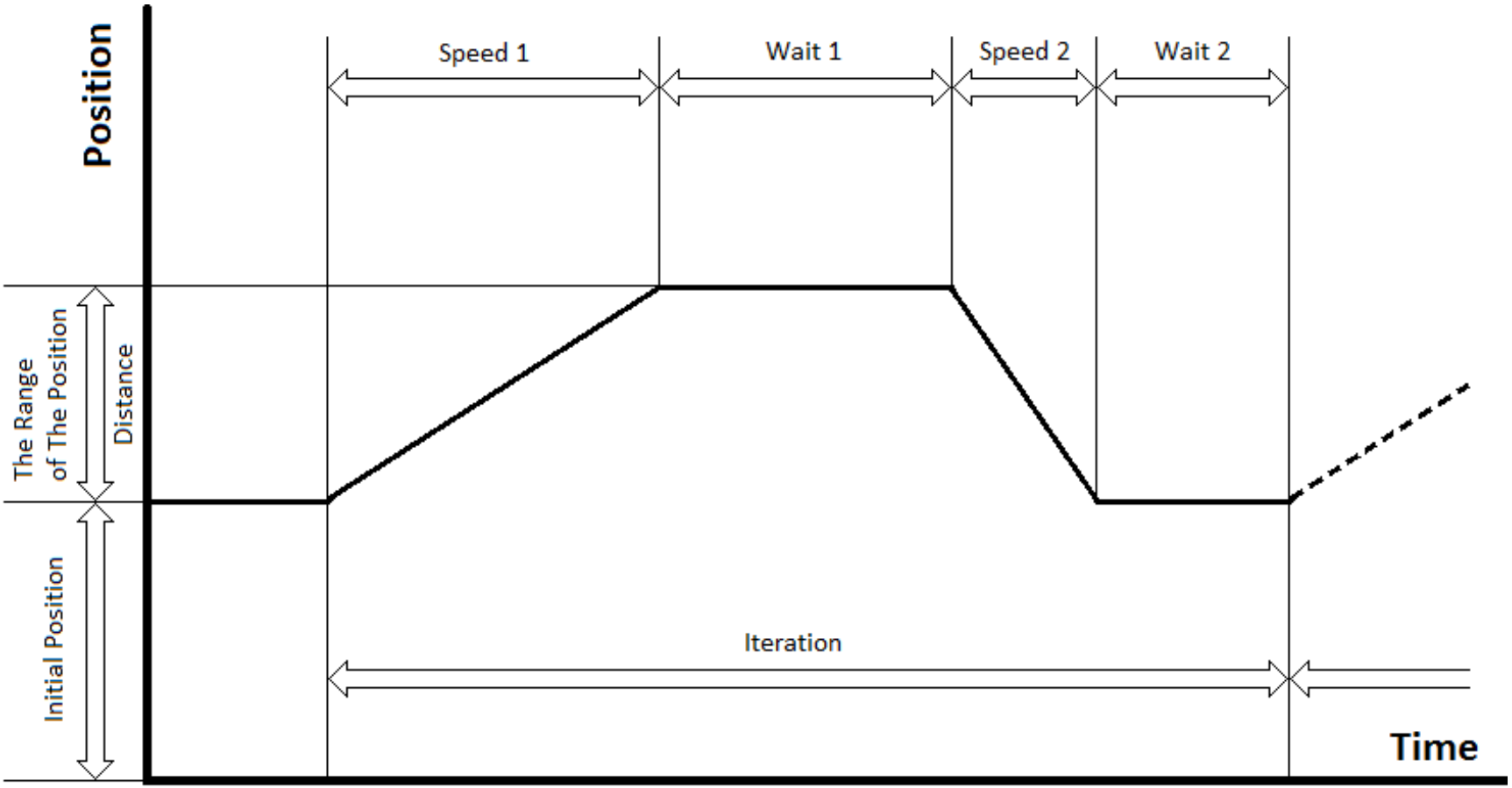
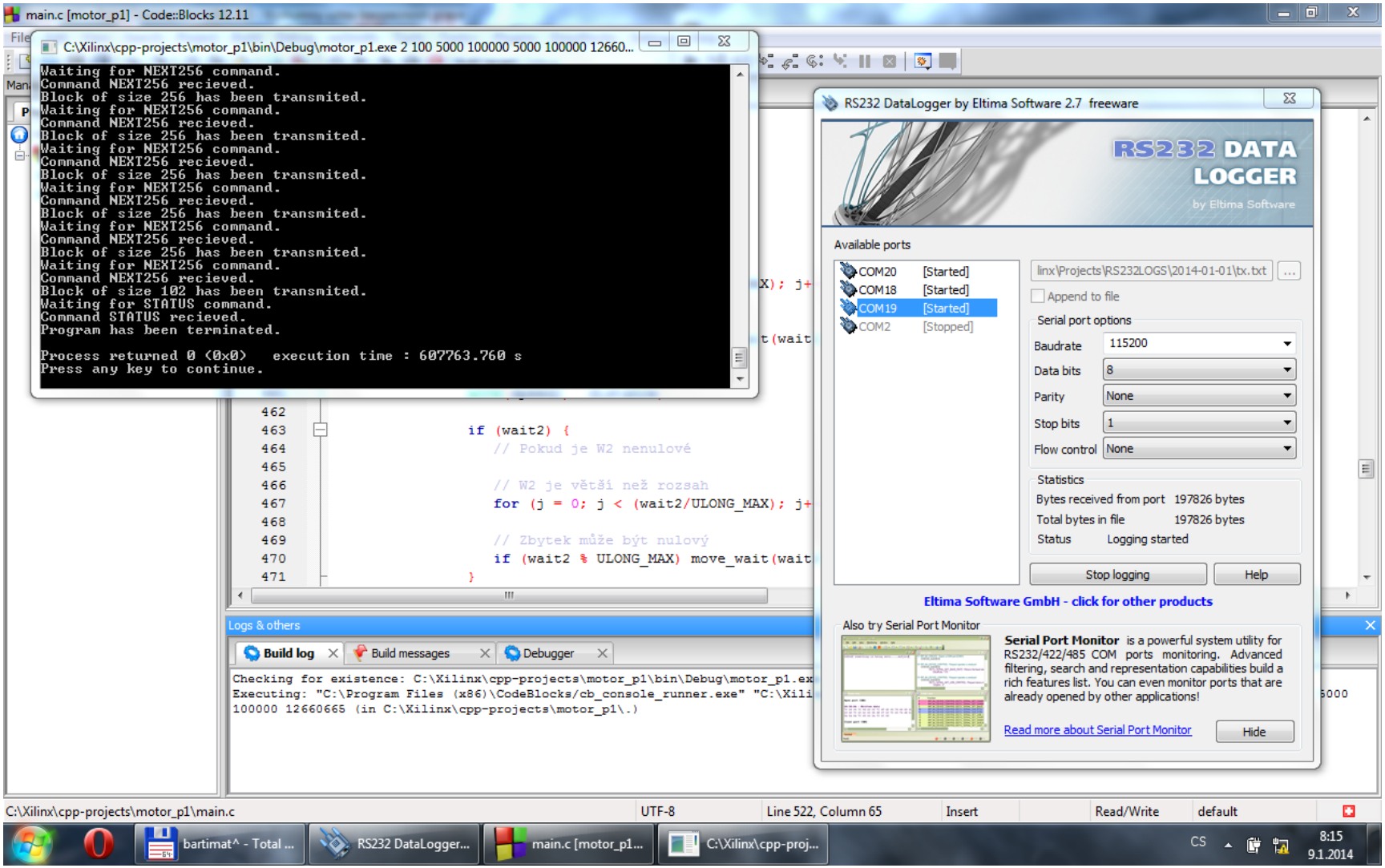


Figure : Performed Movements provided by Testing Control App

Testing Control App — Summary of App



Summary

- I created a new advanced control unit, based on Xilinx FPGA Spartan-3. Design benefits from FPGA architecture.
- Control System is divided into hardware and software parts. Collaboration with MATLAB system is possible.
- Control System is able to perform complex movements thanks to large instruction file.
- Control System is able to handle various events and is designed for being fail–safe.
- The control system and all its parts were exhaustly tested under full operational conditions. One week long test equal to 50 milions of executed instructions.
- The control system actually does not meet EMC directives. New printed circuit board is currently under development to meet the EMC directives.

Visualization of new PCB revision