TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Department of Computer Systems

Artjom Smolov 153154IASM

CONTROL OF MULTITANK SYSTEM USING RASPBERRY PI

Masters's Thesis

Supervisors: Aleksei Tepljakov

Research Scientist,

Eduard Petlenkov

Associate Professor

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutisüsteemide instituut

Artjom Smolov 153154IASM

MULTITANK SÜSTEEMI JUHTIMINE RASPBERRY PI-GA

magistritöö

Juhendajad: Aleksei Tepljakov

Teadur,

Eduard Petlenkov

Dotsent

Tallinn 2017

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Artjom Smolov

12.05.2017

Abstract

The main aim of the given Master's thesis is to develop an application for the microcomputer Raspberry Pi. The application should control Multitank System in real-time through a graphical user interface.

Control of Multitank System is performed on Raspberry Pi. The driver for the Multitank System is implemented in the MATLAB/Simulink environment running on a personal computer. This schematic diagram includes UDP packet sender and receiver blocks. The communication between Multitank System and Raspberry Pi is provided over Wi-Fi network. This allows to develop remote control applications.

It is necessary to choice correct programming language to ensure rapid prototyping of the software application. In our case Python is suitable as a language of fast prototyping. The framework of this research didn't include elaboration of complex algorithms requiring detailed optimization.

Final result of application allows end user control of Multitank System by applying PID controller parameters or applying own conditions. Also application can be further developed in the future.

This thesis is written in English and is 58 pages long, including 9 chapters, 27 figures and 1 table.

Annotatsioon

MULTITANK SÜSTEEMI JUHTIMINE RASPBERRY PI-GA

Raspberry Pi on pangakaardi-suurune ühest trükkplaadist koosnev arvuti, mis oli esialgselt loodud arvutiteaduse õpetamise eesmärgil. Originaalne mudel sai kiiresti populaarseks ja seda hakati kasutama ka mõnedes teistes tööstuslikes rakendustes, näiteks, robootikas.

Raspberry Pi 3 mudel B kuulub kolmandale seadme põlvkonnale. Võrreldes eelnevate mudelitega, on Raspberry Pi 3 võimsam ja on ka täiendatud sisseehitatud side funktsionaalsusega, mis võimaldab kasutada seda otseselt Asjade Interneti osana.

Peamine magistritöö ülesanne on Raspberry Pi 3 rakendamine ühe tööstusliku objekti laboratoorse mudeli juhtimisel. Selleks arendatakse vastavat tarkvara ja luuakse sobivat kasutajaliidest. Juhtimisobjektiks on ühendatud anumate süsteem ja seda juhitakse MATLAB/Simulink keskkonnast läbi andmehõive kaardi. Raspberry Pi suhtleb arvutiga läbi traadita lokaalvõrgu, loeb ühendatud anumate parameetreid ning genereerib juhtimiseks vajalikke käske, mida MATLAB seejärel edastab füüsilisele seadmele reaalajas.

Lõputöö ülesande lahendamiseks valitakse sobivat programmeerimiskeelt ja kohaseid teeke ning arenduskeskkond. Seejärel arendatakse tarkvaralist lahendust. Juhtimisalgoritmiks kasutatakse proportsionaalset-integraalset-diferentsiaalset (PID) regulaatorit. Selle parameetreid valitakse katseliselt ning regulaatori töö valideeritakse eksperimentaalselt.

Väljatöötatud rakendus on saadaval GitHub-st ning seda saab edasi arendada.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 58 leheküljel, 9 peatükki, 27 joonist, 1 tabelit.

Nomenclature

CPU	Central Process Unit
GPU	Graphics processing unit
GUI	Graphical User Interface
IDE	Integrated Development Environment
IP	Internet Protocol
LED	Light Emitting Diode
OS	Operating System
RAM	Random Access Memory
SoC	System On Chip
SSH	Secure Shell
UDP	User Datagram Protocol
VNC	Virtual Network Computing
Wi-Fi	Wireless Local Area Networking

Contents

1 Introduction **12**

1.1 Problem Statement 12

1.2 Objectives and Motivation 13

1.3 Thesis Outline 13

2 Raspberry Pi 3 **15**

2.1 Overview 15

2.2 Hardware 16

2.3 Application 17

3 MultiTank System **18**

3.1 System Overview 18

4 Working System **20**

4.1 Work Schematic Diagram 20

4.2 Matlab Diagram 20

4.3 Matlab Network Settings 21

5 Software Development **25**

5.1 Python 25

5.2 SIP 26

5.3 PyQt 26

5.4 Libraries 29

5.5 PyCharm 32

5.6 Qt Creator 33

6 Graphical User Interface **36**

6.1 Main Window 36

6.2 Basic Window 37

6.3 PID Window 39

7 PID Controller **41**

7.1 PID configuration 41

8 Experimental Verification of Implementation **45**

9 Post Development **48**

Conclusions **49**

Bibliography **50**

A Source Code **53**

B PyQt4 installation on macOS and Raspbian **54**

C Remote communication with Raspberry Pi **57**

D Conversion from UI to Py **58**

List of Figures

2.1 Raspberry Pi 3 board [1] 16

2.2 Raspbian OS with PIXEL [2] 17

3.1 Multitank System 19

4.1 Workflow scheme 20

4.2 Simulink diagram implementing device communication 21

4.3 Packet input block 23

4.4 Packet output block 24

5.1 Qbject hierarchy 28

5.2 Object ownership 28

5.3 PID controller 30

5.4 Matplotlib graph 31

5.5 PyCharm 32

5.6 Qt Creator 35

6.1 Main Window 36

6.2 Not implemented message 37

6.3 Decision message 37

6.4 About message 37

6.5 Socket statement message box 38

6.6 Window inheritance 39

6.7 Basic Window 40

6.8 PID Window 40

7.1 Closed loop PID control of a single input, single output system 41

7.2 Level 1 with 15 Set-point 43

7.3 Level 1 with 10 Set-point 44

8.1 Experimental schematic diagram 45

8.2 Comparing of implementation result 47

B.1 Raspbian memory error during pip installation 56

List of Tables

2.1 Comparison of Raspberry Pi models 16

Chapter 1

Introduction

1.1 Problem Statement

Raspberry Pi is a credit card-sized computer, which has been developed in response to a lack of computer-literate young people and to make computing more widespread in classrooms, home and the Third World. It is the perfect machine for making projects and to learn programming or electronics. It is very portable and can connect to real world objects. For instance, Raspberry Pi could be low cost and flexible alternatives to the usual industrial devices for adding remote control and monitoring ability to small industrial systems. Raspberry PI is very flexible for programming. It supports many programming languages like Python, HTML5, Jquery, Javascript, Java, Perl, Erlang and C/C++. In this case, customizable of programming languages could find easy adaptation to the existing systems and can offer benefits to the industrial world.

The Multitank laboratory system is the perfect tool for researching and teaching automatic control at technical universities. One can perform real-time experiments. The manual valves can be used to simulate leaks and to develop fault-detection strategies.

The Multitank System relates to liquid level control problems commonly occurring in industrial storage tanks. For example, steel producing companies around the world have repeatedly confirmed that substantial benefits are gained from accurate mould level control in continuous bloom casting [3]. Mould level oscillations tend to stir foreign particles and flux powder into molten metal, resulting in surface defects in the final product.

The goal of the Multitank System design is to study and verify in practice linear and nonlinear control methods. The general objective of the control is to reach and stabilise the level in the tanks by an adjustment of the pump operation or/and valves settings. This control problem can be solved by a number of level control strategies ranging from PID to adaptive and fuzzy logic controls.

Python is a general-purpose and popular programming language [4]. It can be used for everything from web development to software development and scientific applications.

1.2 Objectives and Motivation

The main goal of this thesis is to develop an application for controlling Multitank System object. The main application should provide GUI, which allows change parameters of Multitank System. These parameters include motor power, the states of three automatic valves and applying PID controller setting for first level of Multitank System. Second goal is to choose suitable programming language for an apllication. We aim to present GUI on the limited hardware device. Right programming language gives oppurtunity to not overload CPU, RAM and GPU of Raspberry Pi. Also the application is developed on brand new Raspberry Pi 3 Model B and it has own hardware limitations. The finished work gives an overview about GUI process and controlling object at the same time on Raspberry Pi. After research of similar thesis in Department of Computer Control we found one similar work. It was written by Igor Petrov and thesis topic name „Raspberry Pi based System for Visual Detection of Fluid Level“ [5].

1.3 Thesis Outline

Chapter 2 gives an overview of the Raspberry Pi 3. The reader learns, what is new in Raspberry Pi 3 model B.

In Chapter 3 the reader is presented with an overview of the Multitank system.

Chapter 4 focus reader on Working Schematic diagram, which introduce system workflow.

Chapter 5 starts with an overview of Software Developement.

Chapter 6 introduces the reader to the GUI.

Chapter 7 gives an overview of the PID Controller configuration.

In Chapter 8 the reader is presented with an overview of comparing of implementation.

In Chapter 9 the reader gets familiar with post developement.

Chapter 2

Raspberry Pi 3

2.1 Overview

Raspberry Pi is a credit card-sized computer developed by the Raspberry Pi Foundation and initially released in February 2012. It has been since applied to different problems, such as smart home automation [6], wireless sensoring network [7], portable e-health monitoring system [8], recognition for road sign [9], prevention and detection of SQL injection [10], linear algebra computations [11], environmental monitoring [12] and data logging [13].

The Raspberry Pi 3 Model B is a new product from Raspberry Pi Foundation. It came out in February 2016 [1] and form factor is same as Raspberry Pi 2 Model B. The small visual difference in the board is that the power and disk LEDs have been moved to the other side.

The main changes was on hardware:

Network: Broadcom BCM43438: chip provides 2.4GHz 802.11n wireless LAN, Bluetooth Low Energy, and Bluetooth 4.1 Classic radio support.

CPU: Broadcom BCM2837 SoC includes four high-performance CPU with processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory.

GPU: Broadcom VideoCore IV 400MHz.

RAM: 1GB LPDDR2 (900 MHz).

Storage: microSD.

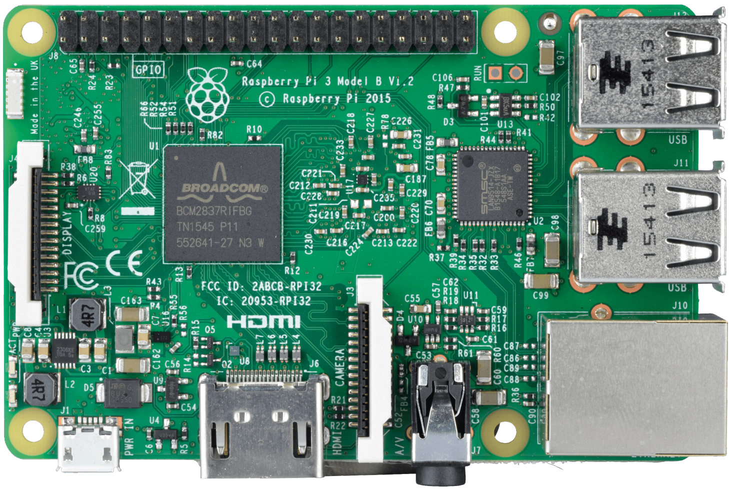

Figure 2.1: Raspberry Pi 3 board [1]

2.2 Hardware

The Raspberry Pi 2 has a Quad Cortex A7, 900MHz CPU, the Raspberry Pi 3 has a quad-core 1.2GHz one. New CPU architecture Cortex-A53 upgrade from 32-bit [14] to 64-bit [15], and gets you better boost for your clock speed. However, both models have 1GB SDRAM and same form factor DDR2. SDRAM frequency for Raspberry Pi 2 is 450MHz, the Pi 3 has 900MHz RAM. Both devices are using a fourth-generation VideoCore IV. The chipset on Raspberry Pi 2 is clocked at 250MHz, on Raspberry Pi 3 clock frequency is 400MHz. The currently requested price for both devices is the same settling at about $35, but there are noticeable differences in performance. A comparison of key parameters of these two boards is provided in Table 2.1.

Table 2.1: Comparison of Raspberry Pi models

	Raspberry PI 2 Model B	**Raspberry PI 3 Model B**
Relase date	2015/02/12	2016/02/29
SoC	BCM2836	BCM2837
CPU	Quad Cortex A7, 900MHz	Quad Cortex A53, 1.2GHz
GPU	250 MHz VideoCore IV	400 MHz VideoCore IV
RAM	1GB SDRAM, 450MHz	1GB SDRAM, 990MHz
Instruction Set	ARMv7-A	ARMv8-A
Storage	micro-SD	micro-SD
Ethernet	10/100	10/100
Wi-Fi	-	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI / Headphone
GPIO	40	40
Price	$35	$35

2.3 Application

Since Raspberry Pi comes without pre-installed OS, there is an opportunity to select suitable OS for user needs. Every OS offers own solution. From proposed OS like Raspbian, NOOBS, Ubuntu Mate, Snappy Ubuntu and Windows 10 IoT Core, we prefer Raspbian OS [16]. The Raspbian OS is a version of Linux (Debian) with a full GUI. It is built specifically for the Raspberry Pi and optimized for Raspberry Pi's hardware. It comes with all the software for every task with a device, contains more than 35.000 packages [17]. Supported officially by the Raspberry foundation.

Raspbian with PIXEL (Pi Improved Xwindow Environment Lightweight) is a software update to the desktop environment. It introduces a brand new interface, and is coming with new programs and features.

The new web application is Chromium Browser, which replaces Epiphany as the default web browser. Also it includes new app such as RealVNC Viewer and Server, which allows you to control Raspberry Pi devices remotely.

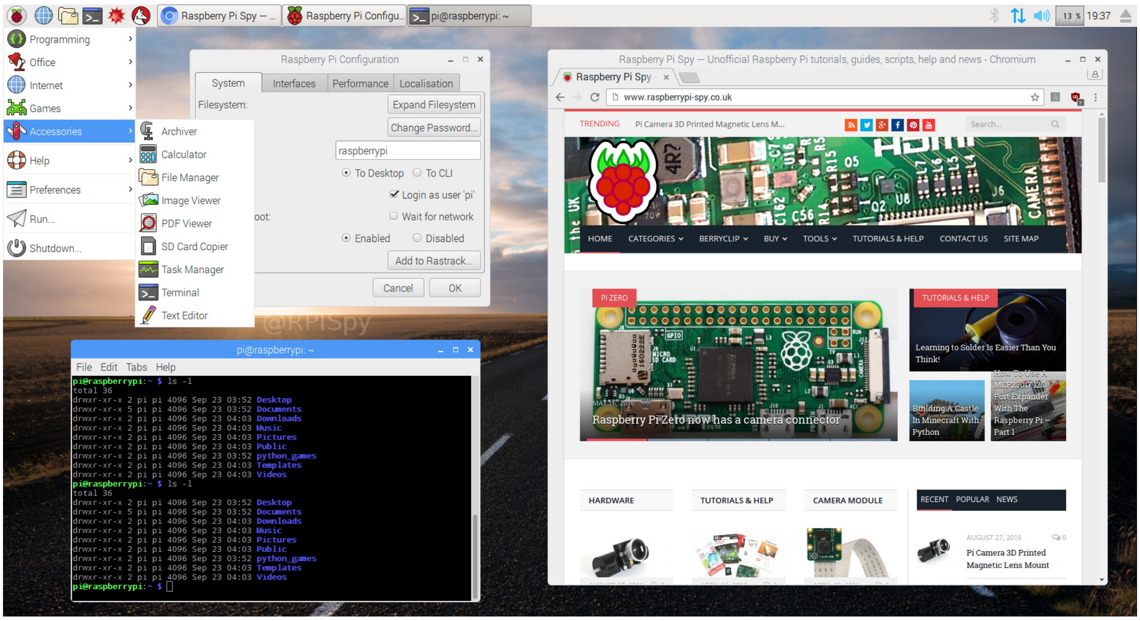

Figure 2.2: Raspbian OS with PIXEL [2]

Chapter 3

MultiTank System

3.1 System Overview

The MultiTank System is an INTECO product [18]. Basically, it illustrates hydroelectricity water flow in miniature. For example the Multitank System relates to liquid level control problems commonly occurring in industrial storage tanks. It comprises a number of separate tanks fitted with drain valves. Two of the tanks have varying cross sections. These introduce nonlinearities into the system. A variable speed pump is used to fill the upper tank. The liquid outflows the tanks due to gravity. The tank valves act as flow resistors. The area ratio of the valves is controlled and can be used to vary the outflow characteristic. Each tank is equipped with a level sensor based on hydraulic pressure measurement [3].

The Multitank System has been designed to operate with an external, PC-based digital controller. The control computer communicates with the level sensors, valves and pump by a dedicated I/O board and the power interface. The I/O board is controlled by the real-time software which operates in MATLAB/Simulink environment.

The goal of the Multitank System design is to study and verify in practice linear and nonlinear control methods. The general objective of the control is to reach and stabilise the level in the tanks by an adjustment of the pump operation or/and valves settings. This control problem can be solved by a number of level control strategies ranging from PID to adaptive and fuzzy logic controls [19].

Hardware:

- 3 tanks made of acrylic glass.
- 2 controlled valves.
- 3 manual valves.
- pump: variable flow, driven by 12 V DC motor.
- 3 level sensors, piezoresistive.
- power interface.
- RT-DAC I/O internal PCI or external USB board.

Dimesions: 350x550x1750 mm.

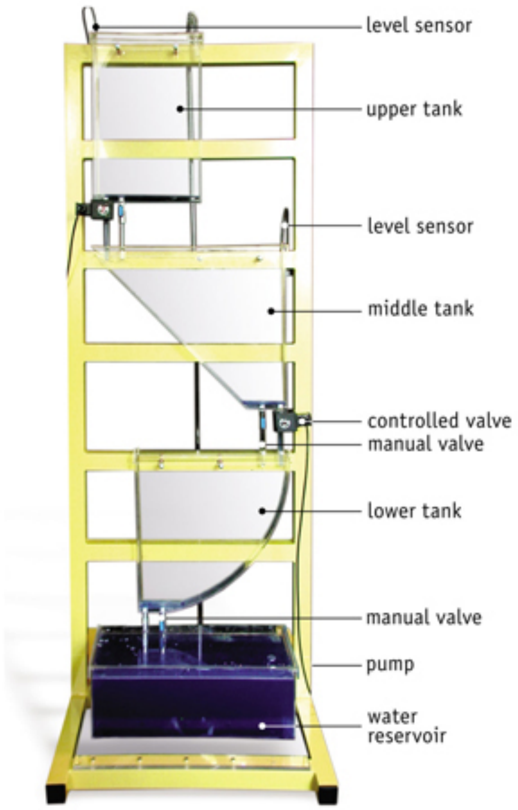

Figure 3.1: Multitank System

Chapter 4

Working System

4.1 Work Schematic Diagram

The working schematic diagram can be seen in Fig. 4.1. Control algorithm is implemented on Raspberry Pi, which is connected to the laboratory computer by Wi-Fi network. Raspberry Pi generates and sends the control signals to the laboratory computer where they are further processed in the MATLAB/Simulink environment. For a transport layer protocol both devices are using UDP packet data. The main task of the laboratory computer is to accept data from the Raspberry Pi and provide signal through data acquisition board for the Multitank System.

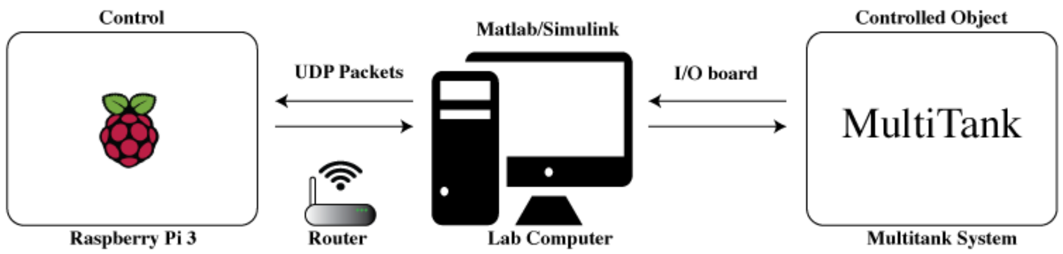

Figure 4.1: Workflow scheme

4.2 Matlab Diagram

We can observe the Matlab/Simulink diagram (see Fig 4.2). The main aim of diagram is to provide runtime and simulation. That means we need to give the value in seconds and execute diagram before Raspberry Pi application starts. When execute was successful it starts sending and recieving UDP packets. To get more detailed information from levels, the diagram inludes a scope block.

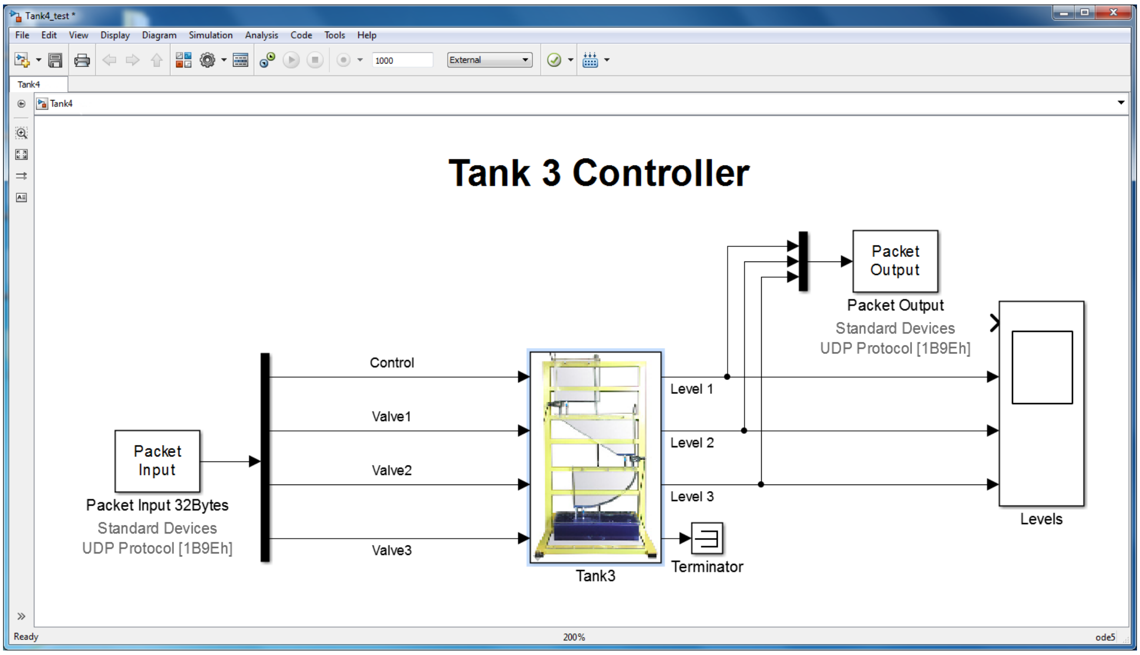

Figure 4.2: Simulink diagram implementing device communication

This diagram consists of:

1. Packet Input Block.

2. Tank Configuration Block.

3. Packet Output Block.

4. Scope Block.

4.3 Matlab Network Settings

The main data type used is *double*. Double precision floating point format is a computer number format that occupies 8 bytes or 64 bits. Main details in diagram are sender block and receiver block. The receiver must accept 32 bytes of data, there each 8 bytes are responsible for own level.

The following serialization of *double* precision floating-point variables is used:

1. First 8 bytes are responsible for motor value.

2. Second 8 bytes are responsible for *level1* value.

3. Third 8 bytes are responsible for *level2* value.

4. Fourth 8 bytes are responsible for *level3* value.

In Fig. 4.3 we will take a look on the main parameters aspect of input block. The first setting is a Board setup, here we need to define IP address, local and remote ports.

After correct Board setup, next list of paramters will be Sample time and Maximum missed ticks. Sample time represents how frequently (in seconds) we want the block to execute and interact with the I/O hardware. Our parameter will be 0.01 this is equal to 100 times per second. Missed ticks is the number of timer ticks that your model can lag behind the real-time kernel, the value 10 will be setted by defaut.

In Input/Output section we declare input packet size in bytes and data type. Each byte range should be multiplied by 4 double data type. Input packet field order is by default - Big Endian.

The provided output block is using same parameters, instead 32 bytes to receive. As we can observe from Fig. 4.2, we require only 24 bytes to send (see Fig 4.4).

The order is:

1. First 8 bytes are responsible for *level1* output value.

2. Second 8 bytes are responsible for *level2* output value.

3. Third 8 bytes are responsible for *level3* output value.

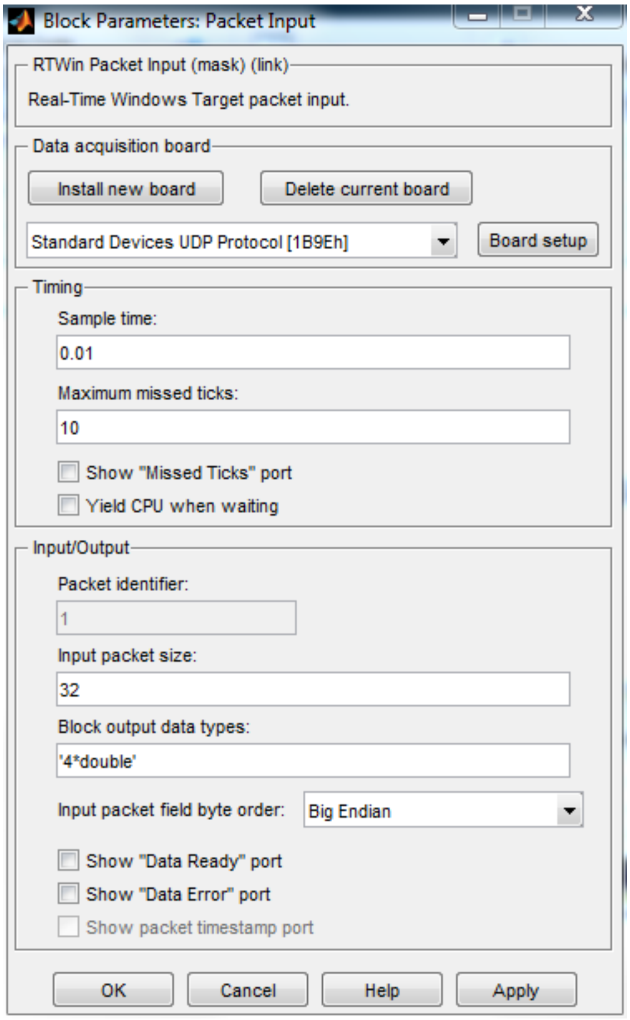

Figure 4.3: Packet input block

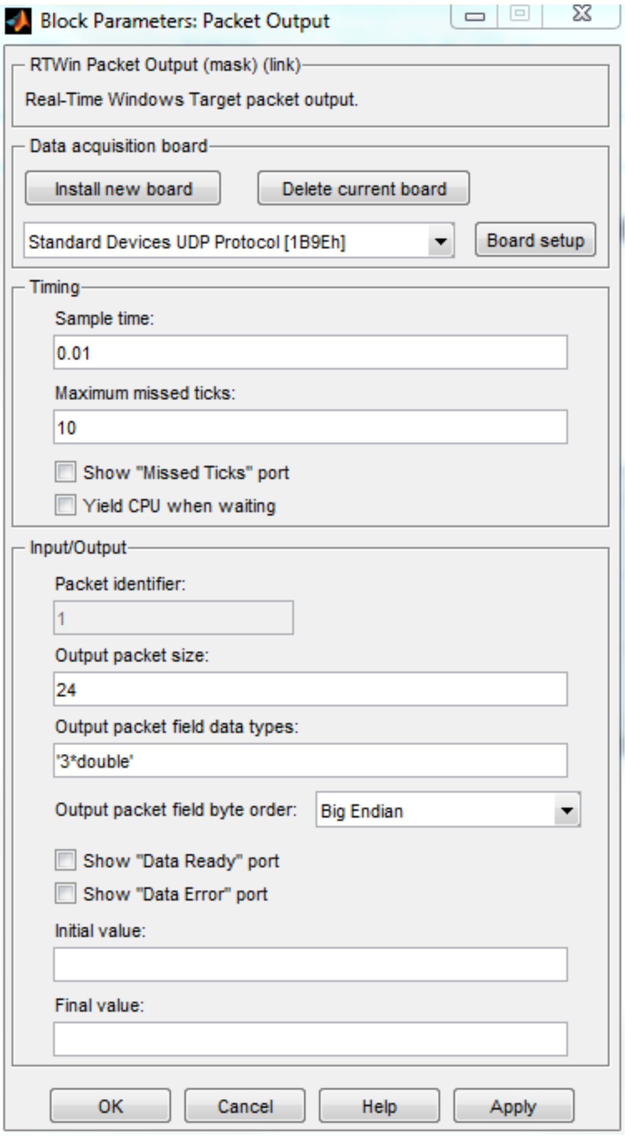

Figure 4.4: Packet output block

Chapter 5

Software Development

5.1 Python

Python was conceived in the late 1980 [20], it can be used on any OS where a Python interpreter is available. The code can be written once and run on almost any computer without needing to change the program. It can be used for games, web applications, network servers, scientific computing, media tools, application scripting and just about anything else you might create on a computer. It is used daily in the operations of the Google search engine, the video-sharing website YouTube and the NASA corporation [21], [22].

Advantages:

- Easy to read code.
- Object Oriented Programming.
- Large number of libraries for Python.
- Open source.
- Great community.

Drawbacks:

- Python is slower than compiled languages.

- Python is not a very good language for mobile development.
- Lack of true multiprocessor support

At the moment Python supports both 2 and 3 versions. The version 2.7.13 is lates stable version, which will be supported up to 2020 [23]. For this thesis was chosen Python 2.7.13, this decion came from the fact that Python 2.7.x has more relevant supporting libraries.

5.2 SIP

SIP is a tool that makes it very easy to create Python bindings for C and C++ libraries. It was originally developed to create PyQt, the Python bindings for the Qt toolkit, but can be used to create bindings for any C or C++ library.

SIP comprises a code generator and a Python module. The code generator processes a set of specification files and generates C or C++ code which is then compiled to create the bindings extension module. The SIP Python module provides support functions to the automatically generated code.

The specification files contain a description of the interface of the C or C++ library, i.e. the classes, methods, functions and variables. The format of a specification file is almost identical to a C or C++ header file, so much so that the easiest way of creating a specification file is to edit the corresponding header file. SIP makes it easy to exploit existing C or C++ libraries in a productive interpretive programming environment [24].

5.3 PyQt

PyQt is a toolkit for creating GUI applications and was developed by Riverbank [25]. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI libraries. Qt is a cross platform development framework written in C++ and developed by Trolltech , but now it is owned by Qt Company [26]. It can be used in several programming languages through bindings with Java, Ruby, Perl and Python (PyQt). PyQt API is a set of modules containing a large number of classes and functions. It has 400+

classes and 6000 functions and methods. These classes are defined in more than 20 modules, which will be presented below. PyQt is compatible with all the popular operating systems including Windows, Linux, and Mac OS. Before you can build PyQt you must have already built and installed **SIP**. PyQt combines all the advantages of Qt and Python.

Frequently used API modules include:

- QtCore — Core non-GUI classes used by other modules.
- QtGui — Graphical user interface components.
- QtMultimedia — Classes for low-level multimedia programming.
- QtNetwork — Classes for network programming.
- QtOpenGL — OpenGL support classes.
- QtScript — Classes for evaluating Qt Scripts.
- QtSql — Classes for database integration using SQL.
- QtSvg — Classes for displaying the contents of SVG files.
- QtWebKit — Classes for rendering and editing HTML.
- QtXml — Classes for handling XML.
- QtAssistant — Support for online help.
- QtDesigner — Classes for extending Qt Designer.

QtCore module contains non-GUI functionality for working with the OS file system. QtGui module contains all the graphical controls. In this thesis we used two modules: QtCore and QtGui.

The QObject class is the base class of all Qt objects. It is the heart of the Qt Object Model. The central feature in this model is a very powerful mechanism for seamless object communication called *signals* and *slots*.

QApplication class manages the main settings and control flow of a GUI application. It contains main event loop inside which events generated by window elements and other sources are processed and dispatched. It also handles system-wide and application-wide settings.

QWidget class derived from QObject classes is the base class for all user interface objects (see Fig 5.1). QDialog and QFrame classes are also derived from QWidget class. They have their own sub-class system.

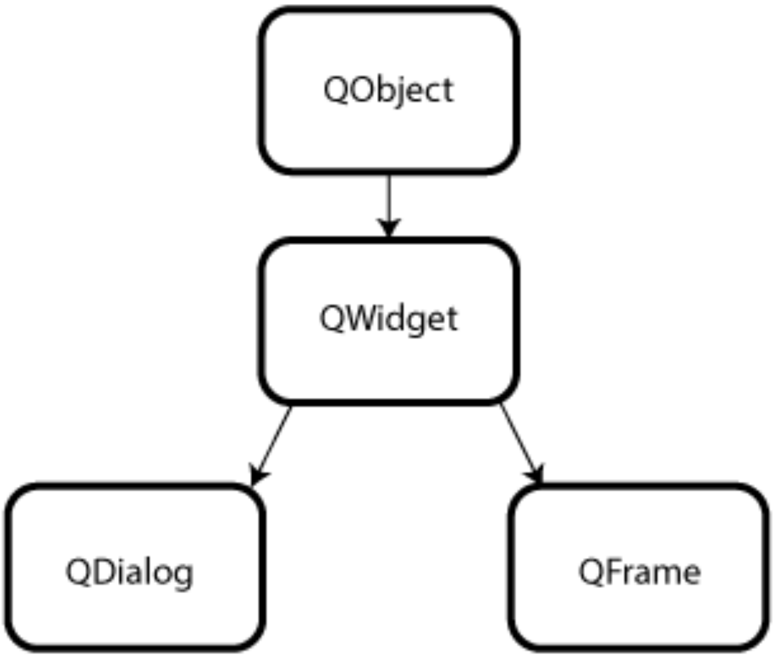

Figure 5.1: Qbject hierarchy

PyQt uses the parent–child ownership (see Fig 5.2). If a parent model is deleted, all its children will be automatically deleted [27].

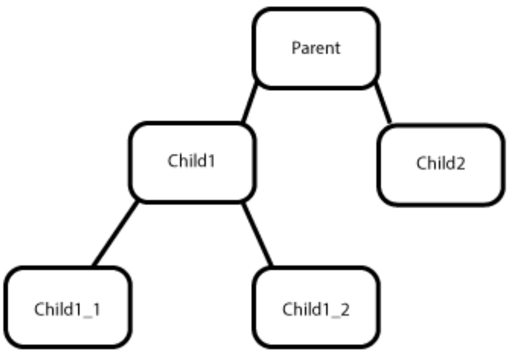

Figure 5.2: Object ownership

At the time of writing the thesis, the latest version of PyQt is PyQt5 (v5.8.2). This thesis work was developed on the PyQt4 (v4.12). The decision was made because of compatibility versions with Python 2.7.13.

5.4 Libraries

This section will introduce reader to the Python libraries, which have been used for developing the applictation.

Sys

This module provides access to some constants, functions, methods variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. It is always available.

Struct

The struct module includes functions for converting between strings of bytes and native Python data types such as numbers and strings.

Threading

This module constructs higher-level threading interfaces on top of the lower level thread module. Using threads allows a program to run multiple operations concurrently in the same process space.

Socket

The socket module exposes the low-level C API for communicating over a network using the BSD socket interface. It includes the socket class, for handling the actual data channel, and functions for network-related tasks such as converting a server's name to an address and formatting data to be sent across the network.

Traceback

This module provides a standard interface to extract, format and print stack traces of Python programs.

CSV

The csv module implements classes to read and write tabular data in CSV format.

Time

This module provides a number of functions to deal with dates and the time within a day. It is a thin layer on top of the C runtime library.

Random

This module implements pseudo-random number generators for various distributions.

PID Ivmech PID Controller is simple implementation of a Proportional-Integral-Derivative (PID) Controller at Python Programming Language [28].

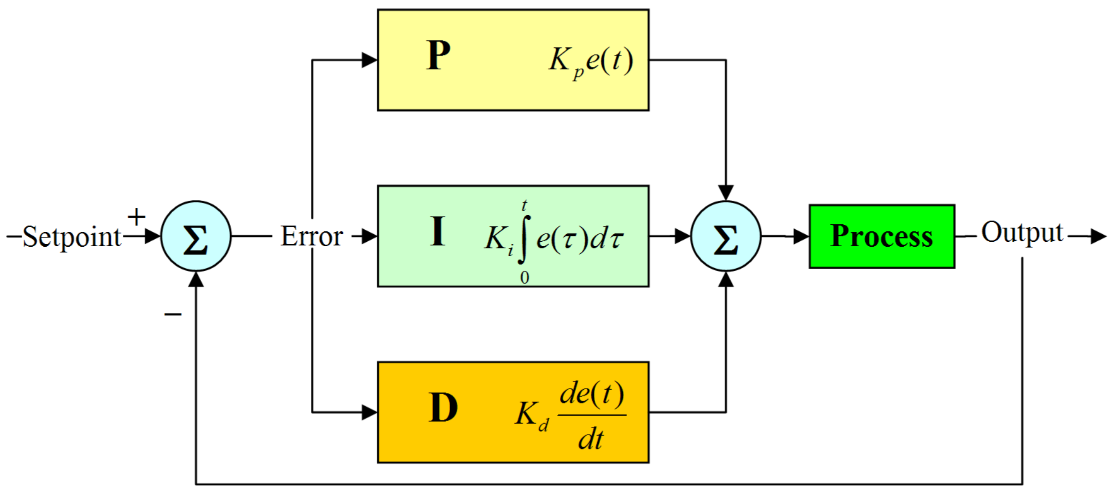

Figure 5.3: PID controller

[Calculates PID value] Calculates PID value for given reference feedback:

$$u(t) = K_p e(t) + K_i \int_0^{\tau} e(t)\mathrm{d}t + K_d \frac{\mathrm{de(t)}}{\mathrm{d}t}. \tag{5.1}$$

Matplotlib

Matplotlib is probably the single most used Python package (plotting library) for 2D-graphics. It provides both a very quick way to visualise data from Python and publication-quality figures in many formats. Matplotlib can be used in Python scripts, MATLAB, Mathematica, web application servers and various graphical user interface toolkits.

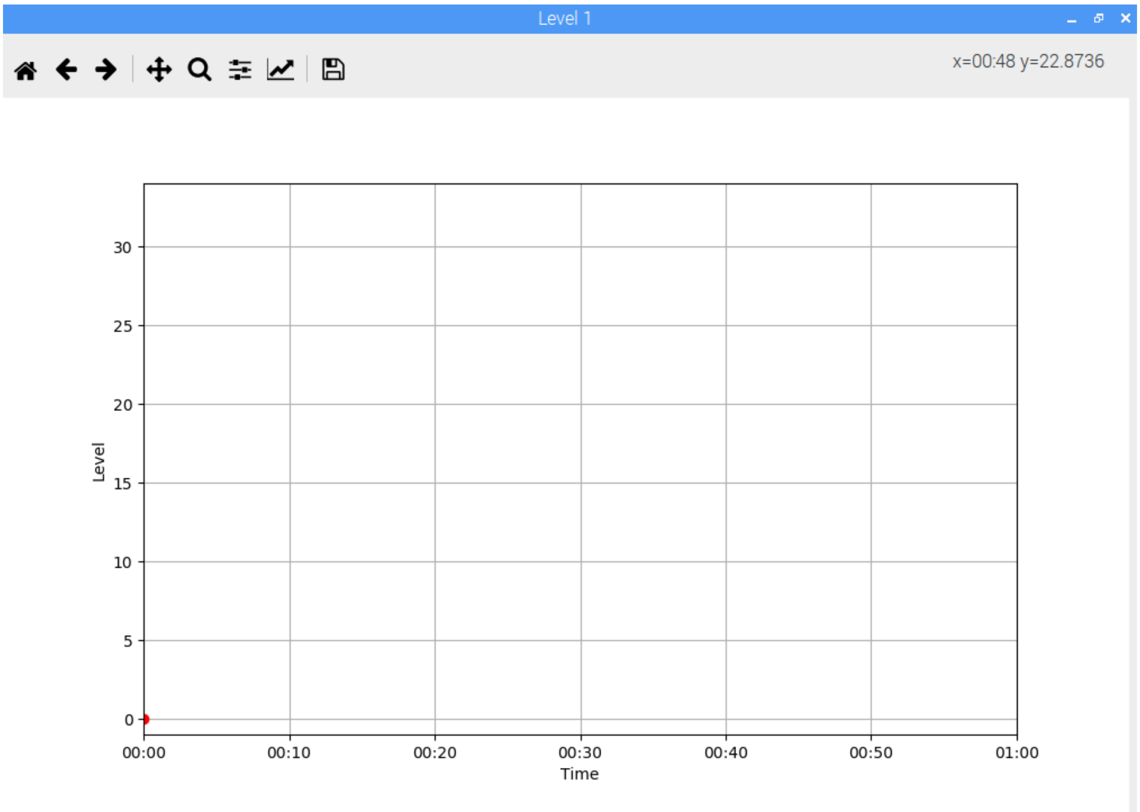

Figure 5.4: Matplotlib graph

NavigationToolbar is an extension for matplotlib. This feature gives interactive navigation. Here is a description of each of the buttons at the top of the toolbar:

1. Home button — always takes you to the first, default view of your data.
2. Back button — navigate back and forth between previously defined views.
3. Forward button — navigate forward and forth between previously defined views.
4. Pan/Zoom button — this button has two modes: pan and zoom.
5. Zoom-to-rectangle button — zooming to selected area.
6. Subplot-configuration button — configure the appearance of the subplot.

7. Edit axes button — axes and curves configuration.

5.5 PyCharm

There are many IDEs available for Python developing like VIM, Wing IDE, Spyder Python and Komodo IDE. After performing analysis of all IDE's and taking into consideration OS version and configuration (development was done on macOS Sierra Version 10.12.4), the best candidate for Python development is PyCharm IDE. First PyCharm was released 2010, but the IDE codebase goes all way back to IntelliJ IDEA which was released as far back as 2001. PyCharm is a child of JetBrains IntelliJ IDEA family of editors and developed by Czech company JetBrains. It supports additional languages JavaScript, HTML, CSS. JetBrains offers free community edition. The community edition is open source.

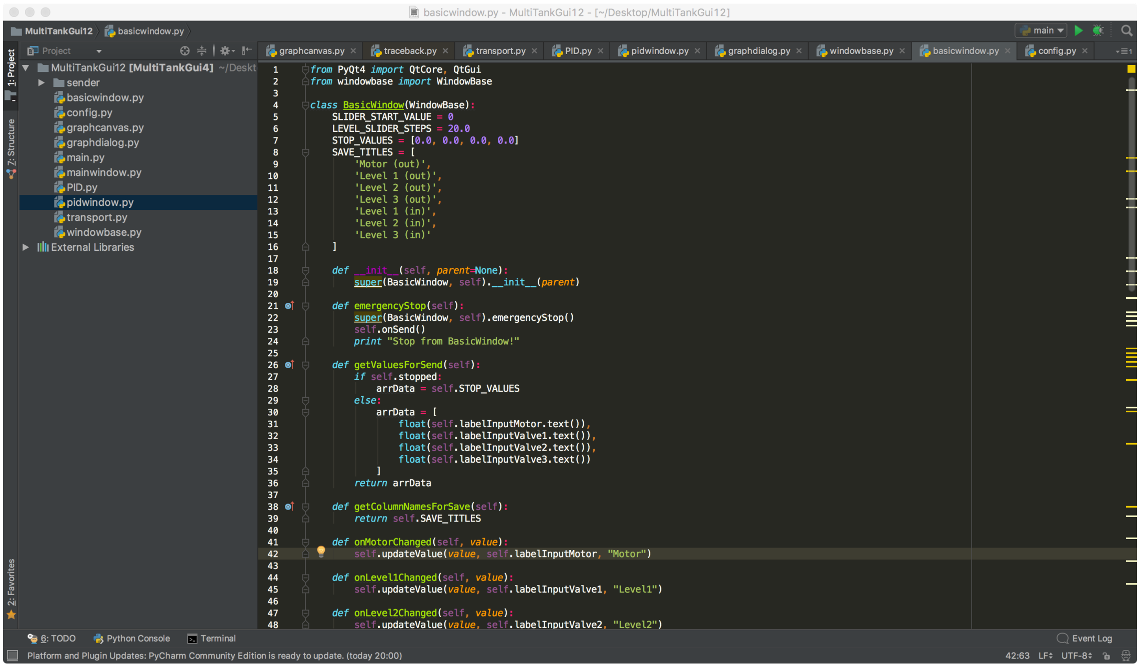

Figure 5.5: PyCharm

PyCharm advantages:

- Integrated debugger.
- Code autocomplete.
- Live package manager.

- Background spellchecking.
- Integrated version control support (Git, SVN).
- Integrated terminal and run window.

PyCharm drawbacks:

- Takes a while to start up IDE.
- Commercial License expensive.

5.6 Qt Creator

Qt Creator is an IDE that provides you with tools to design and develop applications with the Qt application framework. Qt is designed for developing applications and user interfaces once and deploying them to multiple desktop, embedded, and mobile operating systems. Widgets and forms created with Qt Creator integrated seamlessly with programmed code, using Qt's signals and slots mechanism, that lets you easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code. It allows to compose and customize your widgets or dialogs in a "what you see is what you get" manner, and test them using different styles and resolutions.

Managing Projects: One of the major advantages of Qt Creator is that it allows a team of developers to share a project across different development platforms with a common tool for development and debugging.

Designing User Interfaces: Qt Creator provides two integrated visual editors, Qt Quick Designer and Qt Designer. The integration includes project management and code completion.

Coding: Writing, editing, and navigating in source code are core tasks in application development. Therefore, the code editor is one of the key components of Qt Creator. You can use the code editor in the Edit mode.

Building and Running: Qt Creator provides support for building, running, and deploying Qt applications for different target platforms, or using different compilers, debuggers, or

Qt versions. Kits define the tools, device type and other settings to use when building and running your project.

Testing: If you install Qt Creator as part of Qt, the GNU Symbolic Debugger is installed automatically and you should be ready to start debugging after you create a new project. However, you can change the setup to use debugging tools for Windows, for example. You can connect embedded or mobile devices to your development host and debug processes running on the devices.

Publishing: Qt Creator allows you to create installation packages for mobile devices that are suitable for publishing to application stores and other channels.

In order to configure the build of PyQt4 you need to run either the configure-ng.py or the configure.py script. Configure-ng.py is the new configuration script that uses Qt's qmake program to do all the heavy lifting. PyQt configure-ng.py needs both qmake and sip to configure the build process. In order to use PyQt, you need a working installation of Qt. Qt Creator include an **qmake** file which need to be setted as path (see Appendix B). Qmake is a tool that helps simplify the build process for development project across different platforms and qmake generates a makefile based on the information in a project file. It also automatically includes build rules for moc and uic qmake.

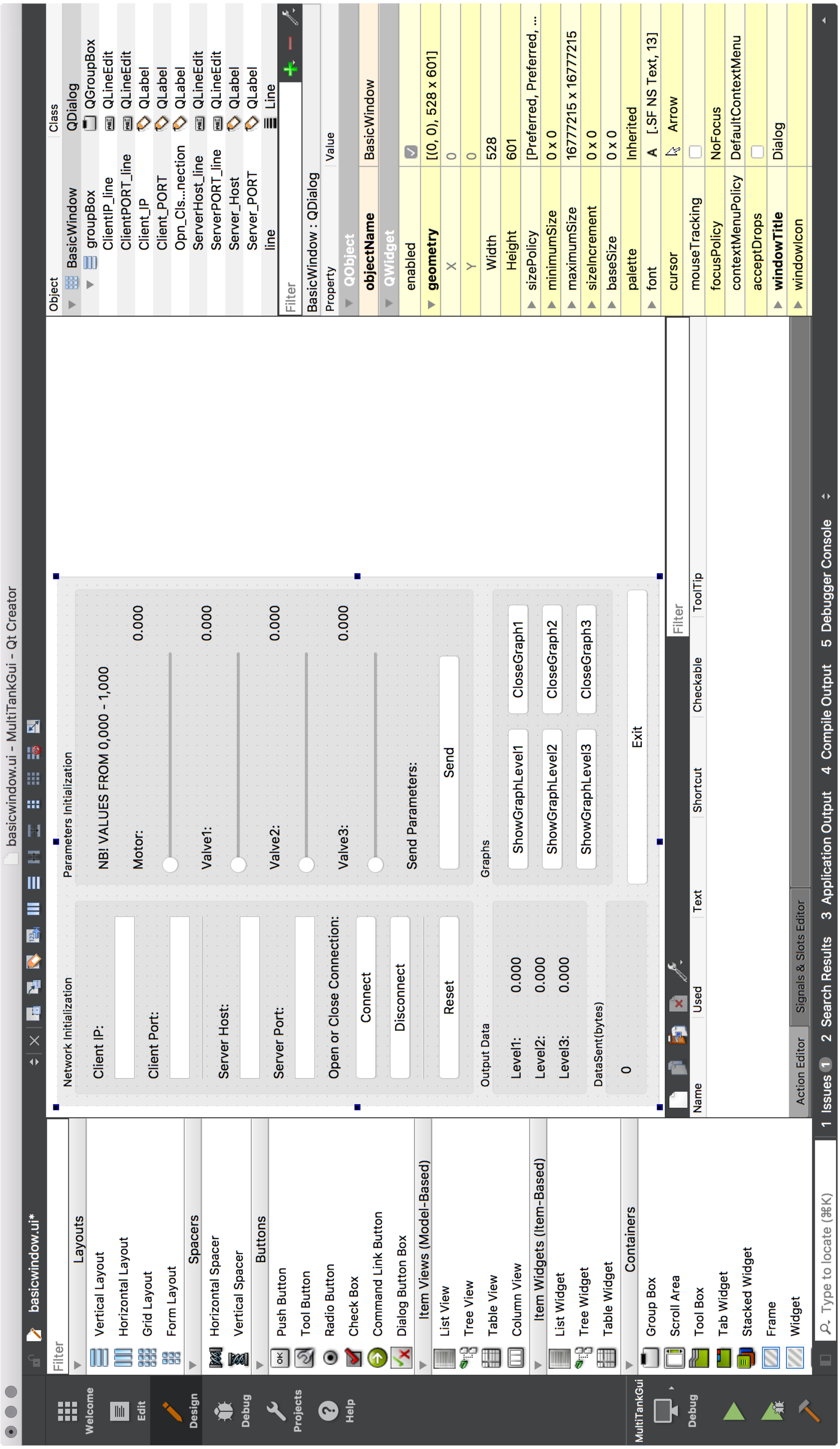

Figure 5.6: Qt Creator

Chapter 6

Graphical User Interface

6.1 Main Window

The GUI application starts from Main Window. It is a small window with couple option buttons and size of Main Window is 176x266 pixels. The grid layout allows resize window without breaking layout formating.

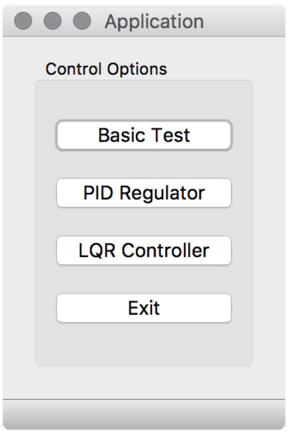

Figure 6.1: Main Window

- The Basic Test button open modal Basic Test window.
- The PID Regulator button open modal PID Controller window.
- The LQR Controller button alert with QMessagebox as Not implemented (see Fig 6.2).
- The Exit button will ask user to confirm a decision for exit option (see Fig 6.3).

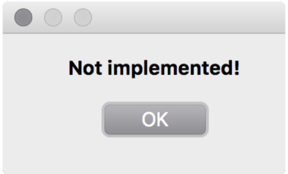

Figure 6.2: Not implemented message

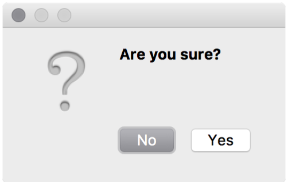

Figure 6.3: Decision message

Also, Qmenubar provides about option. This message will show short information about the author's details (see Fig 6.4).

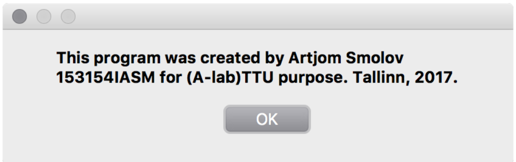

Figure 6.4: About message

6.2 Basic Window

Main feature of the Basic Window (see Fig 6.7) allows user to implement parameters for any situation. Thus, this feature may be helpful at testing or troubleshooting Multitank System. The windows size is 523x601 pixels and contains 5 blocks. Basic Window is including save option to CSV output in menubar.

Network Initialization

This block contain network parameters initialization, socket creation/closing and clear values from DataSent and Output blocks. IP and Port lines have input mask. It protects user from incorrect input, like inserting forbidden letters or symbols. Socket creation/closing will inform user by message box or if socket creation attempt has failed Fig 6.5.

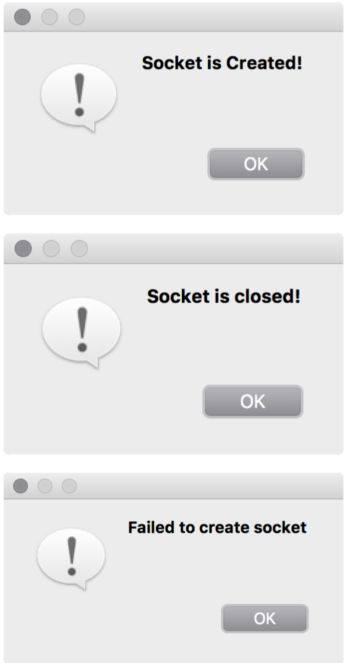

Figure 6.5: Socket statement message box

Output Data

This section provides user about water level in tanks. The output value is shown in centimeters.

DataSent

Here user can observe the value of sent bytes.

Parameters Initialization

This block provides control of Multitank System using 4 sliders. Each slider is responsible for own parameter. The send button will send the current state of all 4 sliders as 32 serialized byte package to the MultiTank System. In emergency situation, the stop button will stop simulation.

Graphs

Graphs can be call from this block, every level have own graph. Stopgraph button will pause data rendering in graph.

6.3 PID Window

PID Window (see Fig 6.8) is same as Basic Window except Parameters Initialization. Basic and PID windows are inherits from WindowBase class as shown in Fig. 6.6. PID Window includes the option to save the output to a CSV file.

The WindowBase class contain 4 blocks:

- Network Initialization.
- Output Data.
- DataSent.
- Graphs.

The Parameters Initialization block is idividual for each window. This inheritance option improves future development, for example if programer wants to create new window like LQR, Fuzzy Logic, Neural Network he/she can inherits from WindowBase class. Only Parameters Initialization is speacial for own window.

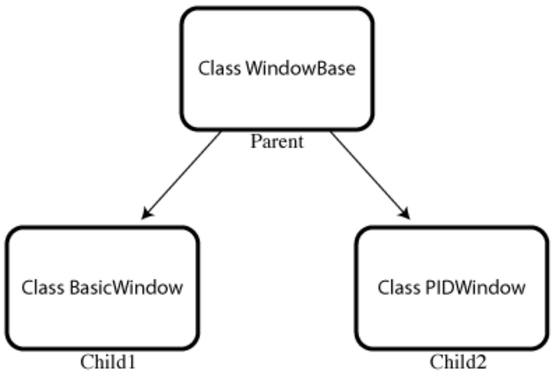

Figure 6.6: Window inheritance

Parameters Initialization

This block provides level choice, set point value, PID parameters, stop button for emergency situation and resume button to continue simulation.

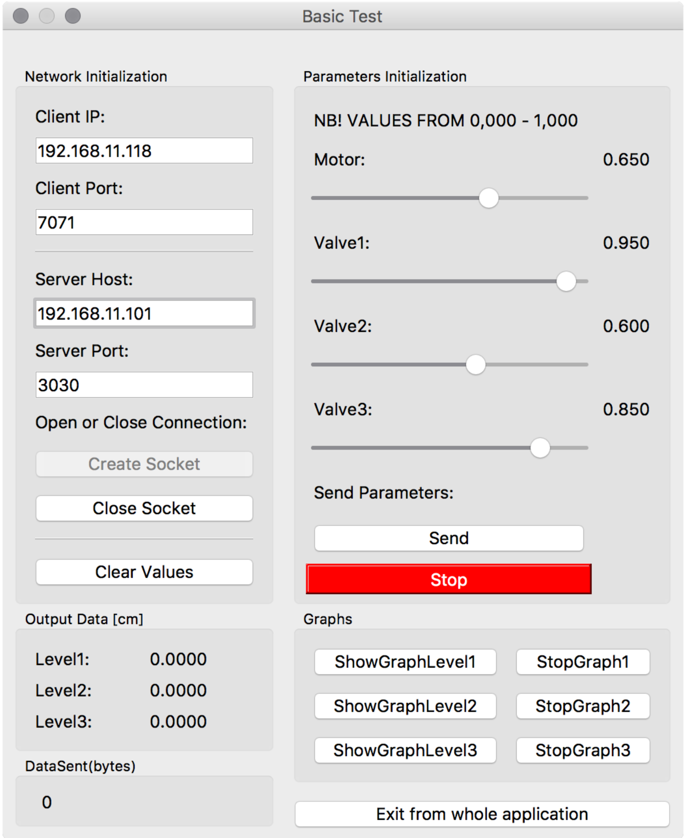

Figure 6.7: Basic Window

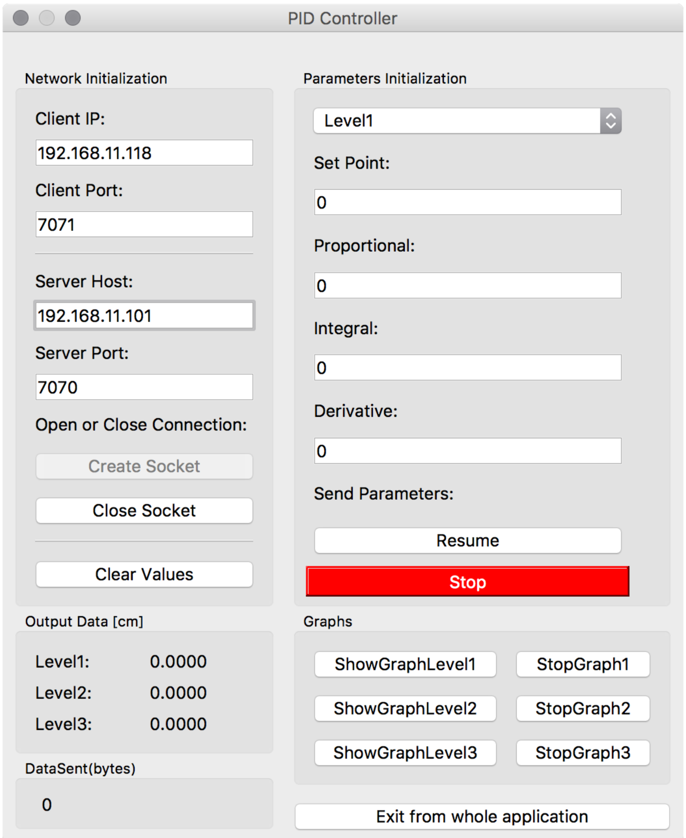

Figure 6.8: PID Window

Chapter 7

PID Controller

7.1 PID configuration

PID stands for Proportional, Integral, Derivative. Cruise control in a car and a themperature adjustment are common examples of how controllers are used to automatically adjust some variable to hold the measurement process variable at the set-point [29]. The set-point is where you would like the measurement to be. Error is defined as the difference between set-point and measurement. With proportional gain, the controller output is proportional to the error (current time) or a change in measurement. With integral action, the controller output is proportional to the amount of time the error is present. With derivative action, the controller output is proportional to the rate of change of the measurement or error. The controller output is calculated by the rate of change of the measurement with time. By default the Proportional, Integral, Derivative and Setpoint values by default is 0. User is allowed to set only these parameters.

Figure 7.1: Closed loop PID control of a single input, single output system

In Fig. 7.1 presented PID diagram of control workflow. It calculates error which provide feedback (output). We are interested for height for example liquid level, where calulated in centimeters. The Multitank System is nonlinear system and disturbances is normal part of the system. Overshot is unavoidable for a nonlinear system. For experiment was proposed 2 set points, 10 and 15. Also, it tested on *level1*.

Set-point 15

For set-point 15 was used: $K_p = 8, K_i = 4,\ K_d = 0$.

The goal is to track a reference value without overshoot during the transient phase. These PI settings are suitable for set-point 15. In Fig. 7.2 the dynamics of the output level under closed loop PID control are shown.

Set-point 10

For set-point 10 was used: $K_p = 6,\ K_i = 4,\ K_d = 0$.

As shown in Fig. 7.3 the output stabilizes at the set point value within approximately 30 seconds.

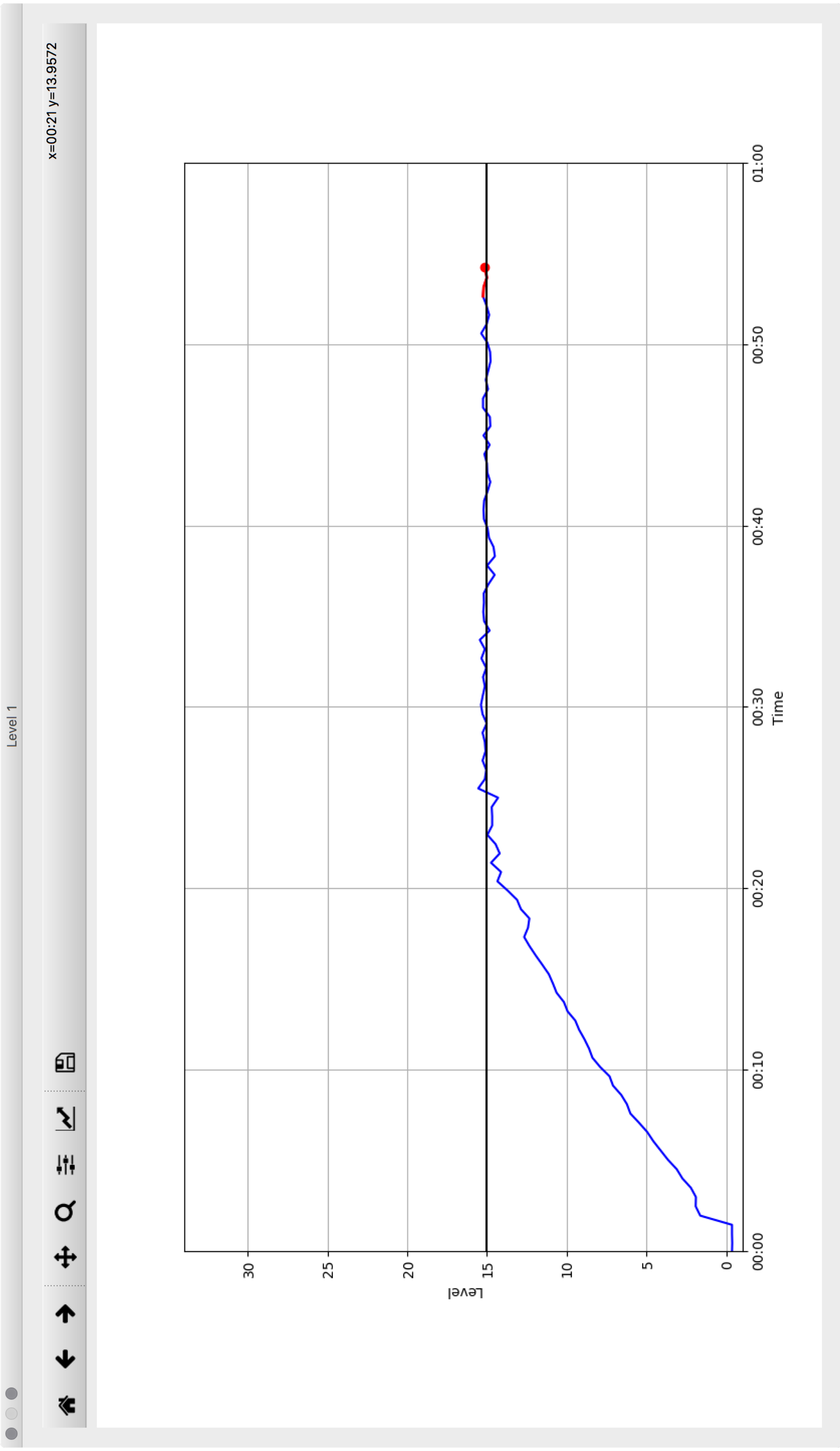

Figure 7.2: Level 1 with 15 Set-point

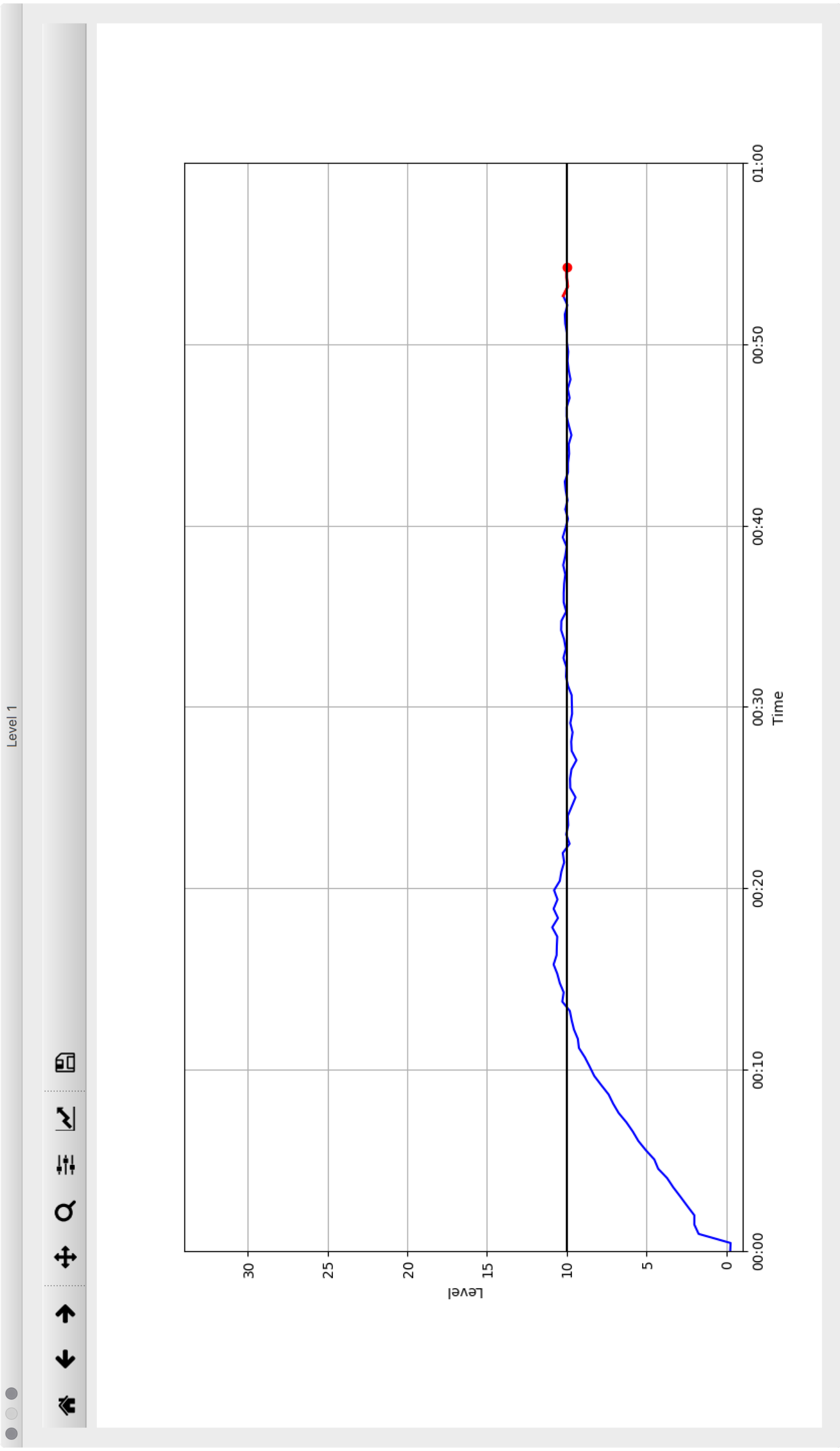

Figure 7.3: Level 1 with 10 Set-point

Chapter 8

Experimental Verification of Implementation

For the purpose of validating the implementation of the control system on Raspberry Pi, a similar control loop was created by means of a Simulink diagram in the MATLAB environment. The same PID controller parameters are used ($K_p = 8$, $K_i = 4$, $K_d = 0$) and the goal is to control the liquid level in the first tank. The set point is chosen as 15 cm, sampling interval is 10ms, and the duration of the experiment is 50s. The sampling interval for the PID controller is different and is 0.5s as it is implemented in the Raspberry Pi application. The corresponding Simulink diagram is shown in Fig 8.1.

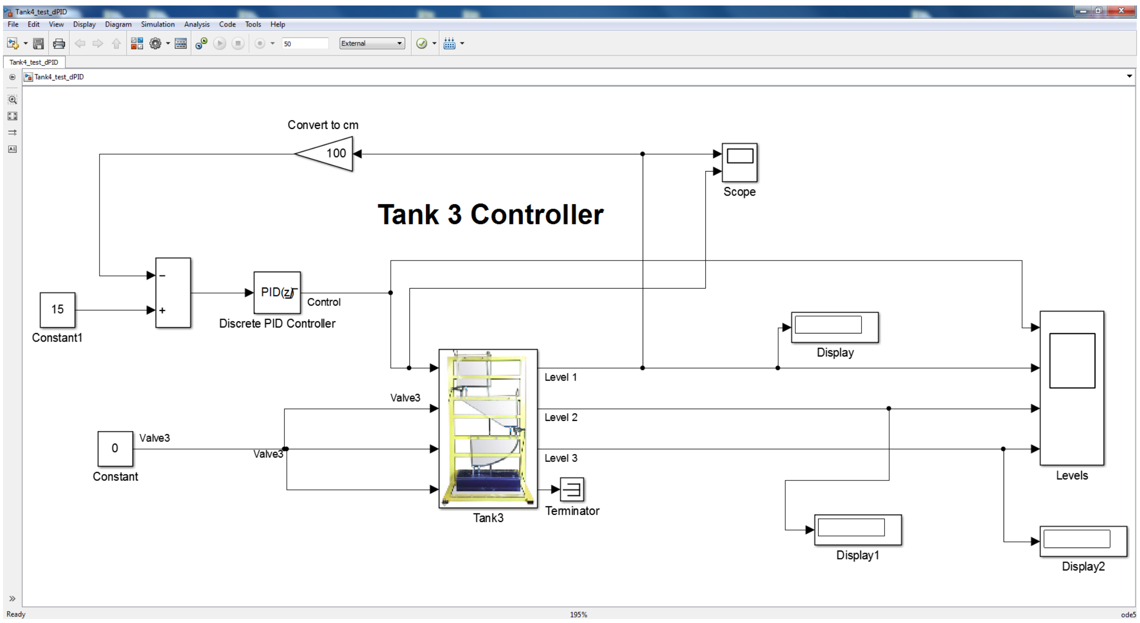

Figure 8.1: Experimental schematic diagram

This diagram consists of:

1. Constant Block.
2. Discrete PID Controller Block.
3. Tank Configuration Block.
4. Subtract Block.
5. Scope Block.
6. Display Block.

The control dynamics are very similar, in both cases damped oscillations can be observed (see Fig 8.2). However, due to difference in MATLAB implementation of the anti-windup mechanism of the PID controller, the amplitude of oscillations in the MATLAB implementation is higher. The control law plot clearly shows that the MATLAB controller generates a control law with a larger amplitude. Yet the feasible range for the real-world pump control signal lies within saturation limits $u_p \in [0,1]$. In both cases this initially leads to the pump motor working at maximum capacity. However, when the output reaches the desired setpoint, the differences in generated control laws are more pronounced, and the Raspberry Pi implementation of the PID control algorithm is clearly superior to the standard clamped anti-windup implementation of the default Simulink Discrete PID controller block since the corresponding control law lies within the saturation boundaries of the pump control signal.

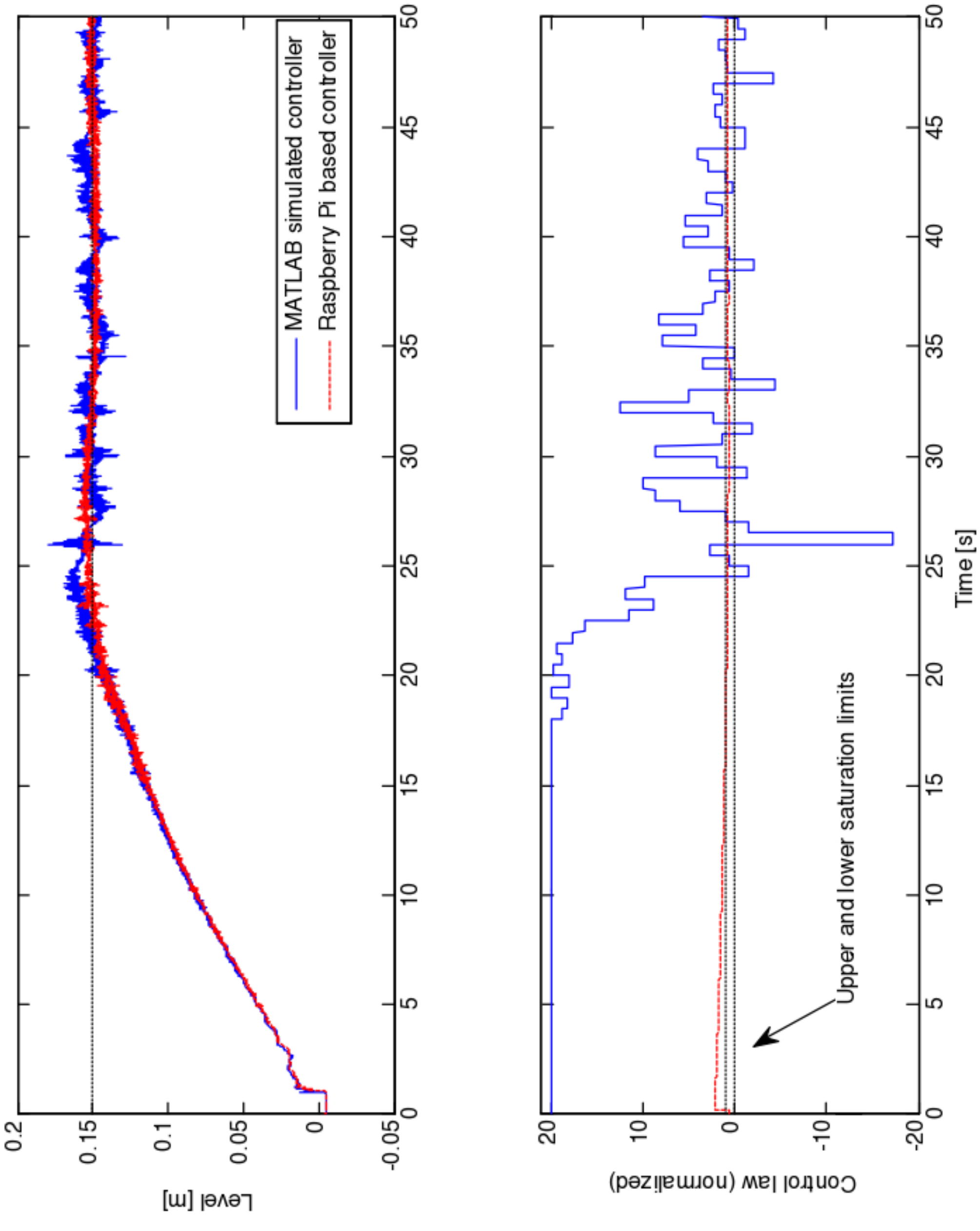

Figure 8.2: Comparing of implementation result

Chapter 9

Post Development

Application can be easily developed in the future. Firstly, preferable re-write application to the Python version 3 and implement new PyQt5 library.

Secondly, additional control schemes can be implemented including Fuzzy logic, LQR or Neural Network.

Thirdly, a mobile application (iOS) could be developed, which allows user to collect data and change parameters. Nowadays a mobile device is more powerful than an office computer, the implemetation of mobile application is great idea.

Finally, a web-based front-end application could be developed. User can access web interface by own login and password to monitoring system process.

The bonus point is to add notification alerts such as email or sms. In case of emergency sistuations user will be always informed by alert.

Conclusions

The goal of this master thesis has been to create an application and implement on Raspberry Pi. It can be viewed as a starting point for implementation of different controllers.

Despite the limited hardware opportunities, the Raspberry Pi is great microcomputer to do different experiments. However there is one important point, the Raspberry Pi OS is running by Raspbian OS based on Linux (Debian). This OS is very limited for outside installations, for example if you need an application you can only install it by "sudo apt-get" command. Thus, there can be problems with cross-OS platform development, because the mismatch of versions that offers package lists from the Raspian OS repositories. Developer must always consider the difference between the versions on the macOS or Windows.

Development of the software applications only in Raspberry Pi require additional monitor (with HDMI output), mouse and keyboard. It is a good opportunity to create small applications which don't need to debug. All IDE for Raspberry Pi is simplified, because it caused by hardware limitation.

The Raspberry Pi showed their best side, it is great credit card-sized computer for the implementation of control algorithms.

Python is perfect programming language for script tasks. For creating GUI is require additional libraries, additional settings. It is fine when development realizing for single OS system, but for cross-OS platform, developer must consider at small details like version, libraries compatibility. For an experiment it is great to rewrite this application on the different programming language like Java + Swing or C++ with Qt.

Bibliography

[1] RASPBERRY PI FOUNDATION. Raspberry Pi model B. Last accessed 2017-01-17. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[2] MATT. (2016) Introducing PIXEL the New Raspbian Desktop. Last accessed 2017-04-20. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[3] INTECO, *Multitank user manual*, 2012.

[4] Most popular coding languages of 2016. Last accessed 2016-12-18. [Online]. Available: http://blog.codeeval.com/codeevalblog/2016/2/2/most-popular-coding-languages-of-2016

[5] I. Petrov, "Raspberry Pi based system for visual detection of fluid level," 2014. [Online]. Available: http://a-lab.ee/edu/theses/defended/437

[6] V. Patchava, H. B. Kandala, and P. R. Babu, "A Smart Home Automation technique with Raspberry Pi using IoT," in *2015 International Conference on Smart Sensors and Systems (IC-SSS)*. IEEE, December 2015, pp. 1 – 4.

[7] S. G. Nikhade, "Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. IEEE, May 2015, pp. 376 – 381.

[8] M. C. Paul, S. Sarkar, M. M. Rahman, S. M. Reza, and M. S. Kaiser, "Low cost and portable patient monitoring system for e-Health services in Bangladesh," in *2016*

International Conference on Computer Communication and Informatics (ICCCI). IEEE, Januar 2016, pp. 1 – 4.

[9] E. Bilgin and S. Robila, "Road sign recognition system on Raspberry Pi," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. IEEE, April 2016, pp. 1 – 5.

[10] S. Djanali, F. Arunanto, B. A. Pratomo, H. Studiawan, and S. G. Nugraha, "SQL injection detection and prevention system with Raspberry Pi honeypot cluster for trapping attacker," in *2014 International Symposium on Technology Management and Emerging Technologies*. IEEE, May 2014, pp. 163 – 166.

[11] N. Kyrtatas, D. G. Spampinato, and M. Püschel, "A basic linear algebra compiler for embedded processors," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*. IEEE Conference Publications, 2015, pp. 1054 – 1059.

[12] V. S. Srinivasan, Saleem, T. Kumar, and D. K. Yasarapu, "Raspberry Pi and ibeacons as environmental data monitors and the potential applications in a growing BigData ecosystem," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, may 2016, pp. 961 – 965.

[13] V. Pasquali, R. Gualtieri, G. D. Alessandro, F. Leccese, and M. Cagnetti, "Experimental in field reliability test for data logger based on Raspberry-Pi for extreme scenarios: A first step versus aerospace applications," in *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*. IEEE, June 2016, pp. 365 – 370.

[14] ARM Cortex-A7. Last accessed 2017-02-22. [Online]. Available: https://en.wikipedia.org/wiki/ARM_Cortex-A7

[15] ARM Cortex-A53. Last accessed 2017-02-22. [Online]. Available: https://en.wikipedia.org/wiki/ARM_Cortex-A53

[16] Raspberry Pi downloads. Last accessed 2017-01-20. [Online]. Available: https://www.raspberrypi.org/downloads/

[17] Raspbian Operating System. Last accessed 2016-12-11. [Online]. Available: https://www.raspbian.org/

[18] INTECO. http://www.inteco.com.pl/. Last accessed 2017-04-30.

[19] Multitank overview. Last accessed 2016-10-04. [Online]. Available: http://www.inteco.com.pl/products/multi-tank/

[20] V. Bill. (2003) The making of Python. Last accessed 2017-02-12. [Online]. Available: http://www.artima.com/intv/python.html

[21] A. Sangster. Organizations Using Python. Last accessed 2017-05-02. [Online]. Available: https://wiki.python.org/moin/OrganizationsUsingPython

[22] Quintagroup. Organizations Using Python. Last accessed 2017-04-26. [Online]. Available: http://quintagroup.com/cms/python/google

[23] A. K. Kuchling. (2017) The future for Python 2.xthe future for Python 2.x. Last accessed 2017-01-21. [Online]. Available: https://docs.python.org/2/whatsnew/2.7.html

[24] Riverbank. (2014) What is SIP? Last accessed 2017-02-03. [Online]. Available: https://www.riverbankcomputing.com/software/sip/intro

[25] Riverbank. Last accessed 2016-12-21. [Online]. Available: https://www.riverbankcomputing.com/news

[26] The Qt company. Last accessed 2016-11-05. [Online]. Available: https://en.wikipedia.org/wiki/The_Qt_Company

[27] M. Summerfield, *Rapid GUI Programming with Python and Qt*, 2009, vol. 550.

[28] Ivmech. (2016) ivpid. Last accessed 2017-03-08. [Online]. Available: https://github.com/ivmech/ivPID

[29] K. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. The Instrumentation, Systems, and Automation Society (ISA), 1995.

Appendix A

Source Code

The application program source code is available at:

"`https://github.com/ArtjomIASM/MultitankGUI`".

The application can be run using the following script in terminal/cmd/command line:

"python main.py" – runs application.

Please, be sure that you are located in same folder as main.py or you can run with adding folder path like:

"python /Folder/main.py".

Appendix B

PyQt4 installation on macOS and Raspbian

This guide will show installation process of PyQt4 on macOS and Raspbian.

macOS

SIP

1. Download SIP (4.19.1) for macOS from official website[1].
2. Unpack package.
3. Locate at SIP folder from package on terminal.
4. From terminal use script "python configure.py".
5. Perform "make" command to create makefile.
6. Perform "sudo make install" command to do installation.
7. Perform "sudo make clean" command to clean all the already compiled object files.
8. In terminal check SIP version by "sip -V", if you get output "4.19.1" then installation is complete.

[1] https://www.riverbankcomputing.com/software/sip/download

PyQt4

1. Download PyQt4 (4.12) for macOS from official website[2].
2. Download Qt Creator (5.7.1 or late version) for macOS from official website[3].
3. Install Qt Creator.
4. Unpack PyQt4.
5. Locate at PyQt4 folder from package in terminal.
6. Use script "python configure-ng.py --qmake /Users/name/Qt/5.7.1/clang_64/bin/qmake" and provide path to the qmake file location.
7. Perform "make" command to create makefile (make processing may take about 10-20 minutes).
8. Perform "sudo make install" command to do installation (installation may take from 10-45 minutes).
9. Perform "sudo make clean" command to clean all the already compiled object files.
10. Install additional packet Matplotlib by using pip installer "pip install matplotlib".
11. In terminal check PyQt4 version by "pyuic4 --version" command, if you get output "Python User Interface Compiler 4.12 for Qt version 5.7.1" then installation is complete.

Raspbian

1. Raspbian is shipped with old version of pip (1.5.3) and it should be upgraded to the new version (9.0.1) by command "pip install pip".
2. Check pip version by command "pip -V", it should be 9.0.1.
3. Installation of PyQt4 is performed by command "sudo apt-get install python-qt4" (SIP is included in package).

[2]https://www.riverbankcomputing.com/software/pyqt/download
[3]http://info.qt.io/download-qt-for-application-development

4. Installation of Qt Designer is performed by command "sudo apt-get install qt-designer".

5. To avoid memory error, use command "pip --no-cache-dir install matplotlib" to get new version of matplotlib (2.0.0).

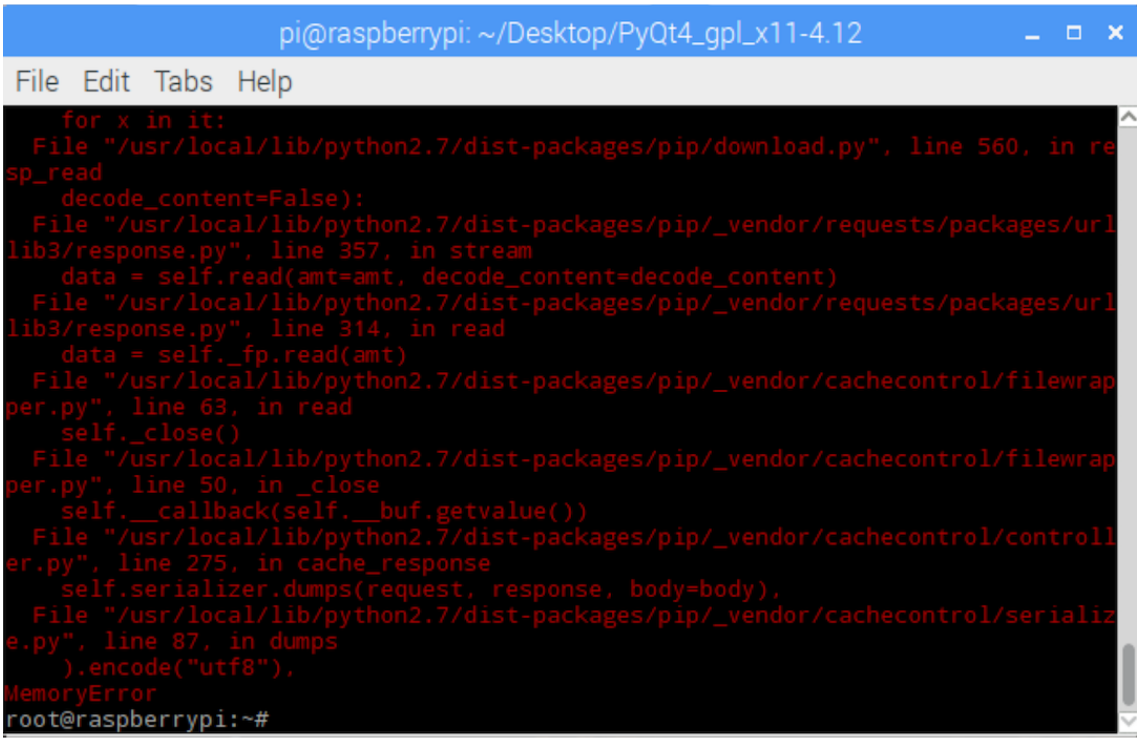

Figure B.1: Raspbian memory error during pip installation

Appendix C

Remote communication with Raspberry Pi

As of the November 2016 release, Raspbian has the SSH server disabled by default. It can be enabled manually from the desktop:

Raspbian

1. Launch Raspberry Pi Configuration from the Preferences menu.
2. Navigate to the Interfaces tab.
3. Select Enabled next to SSH.
4. Hit OK.

macOS

1. Open terminal and write command "ssh pi@192.168.0.111". Pi is the name of Raspberry Pi by default and IP is the address of the Raspberry Pi in network.
2. Run "vncserver" command to activate vnc server.
3. Use any VNC program for connection.

Appendix D

Conversion from UI to Py

This command converts ui file from Qt Creator to Python file format.

"`pyuic4 input.ui -o output.py`" converts ui into py file without executable file.

"`pyuic4 -x input.ui -o output.py`" converts ui into py file with executable file.