Interface and Data Acquisition from Scientific Instruments

Software Requirements Specification 1.0

11-01-17

Rohit Ranjan

Nalla Prajwal Chandra

Bikash Kumar Tudu

Prasad Prashallu

Prepared for

CS 258 Software Engineering

Spring 2017

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 11/01/17 | Version 1.0 | Bikash Tudu | First Revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[Revision History](#_30j0zll)

[1. Introduction](#_org8xbh014a9)

[1.1 Purpose](#_2et92p0)

[1.2 Scope](#_rh37neeqvdw)

[1.3 Definitions, Acronyms, and Abbreviations](#_3dy6vkm)

[1.4 References](#_1t3h5sf)

[2. General Description](#_mgs0kdxp1tm8)

[2.1 Product Perspective](#_j1qqgdlhdtza)

[2.2 Product Functions](#_3rdcrjn)

[2.3 User Characteristics](#_5tjf0lx9knjb)

[2.4 General Constraints](#_lnxbz9)

[2.5 Assumptions and Dependencies](#_5vg1kg3nygwb)

[3. Specific Requirements](#_i9xk2vy78j3r)

[3.1 External Interface Requirements](#_y7948p7u7ixu)

[3.1.1 User Interfaces](#_wrd4h142c15q)

[3.1.2 Hardware Interfaces](#_e27c85n0fwwl)

[3.1.3 Software Interface](#_eoih4fxjh8xn)

[3.1.4 Communications Interfaces](#_16731bovdeqv)

[3.2 Functional Requirements](#_4i7ojhp)

[3.2.1 Automating the controls of CTC100](#_2xcytpi)

[3.2.2 Reading values of various parameters from CTC100](#_yty28oi1j84n)

[3.3 Non-Functional Requirements](#_1uuy5wqwvuj)

[3.3.1 Performance](#_gnqdhkleffja)

[3.3.2 Reliability](#_at5nceetfpkj)

[3.3.3 Availability](#_3amdd6h9k002)

[3.3.4 Security](#_jw4vgq8um7fk)

[3.4 Design Constraints](#_buxxpuxs64wi)

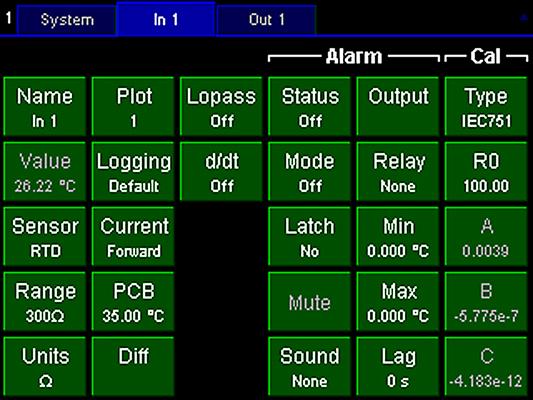
# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description about the ’CTC100 ControlX’ . It explains the purpose and the features of the software, the interfaces of the software, the constraints under which it operates. This document is intended for developers and users of the software.

## 1.2 Scope

’CTC100 ControlX’ is be designed to maximize the user’s productivity by providing tools to assist in automating the process of acquisition of data from the machine(CTC 100) and providing an user friendly interface by which the user can control the machine, which otherwise has to be performed manually. By maximizing the user’s work efficiency and production the software will meet the user’s needs being easy to understand and use.





## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| ***Term*** | ***Definition*** |
| *Field* | *A textbox where input is given.* |
| *Software Requirements Specification* | *A document that completely describes all the functionalities of a proposed system and the constraints under which it must operate. For example, this document.* |
| *User* | *An end user is who uses this software for their personal or research purpose.* |
| *CTC* | *Cryogenic Temperature Controller* |

## 1.4 References

*The following document has been referenced to understand the controls of CTC100.*

*Source :* [*http://www.thinksrs.com/downloads/PDFs/Manuals/CTC100m.pdf*](http://www.thinksrs.com/downloads/PDFs/Manuals/CTC100m.pdf)

# 2. General Description

## 2.1 Product Perspective

This software is specifically designed for Linux environment. The system with Linux installed and CTC100 must share the same local network predominantly an ethernet connection. ’CTC100 ControlX’ is designed to maximize the automation of CTC100. The display of CTC100 Controller has an interactive GUI which helps user to easily configure the CTC100, and control its settings.

## 2.2 Product Functions

**For Output:**

1. Mode on/off

2. Range

3. High and Low limit

4. Input followed

5. Setpoint

6. Final point

7. Interval

8. Tolerance

9. Ramp

10. Hold

**For Input:**

1. Mode on/off

2. Sensor

3. Range

The software enables the user to measure various scientific dimensions at regular intervals of given time specified by the user. It automates the process of taking regular readings.

## 2.3 User Characteristics

## 2.4 General Constraints

This software is specifically designed for Linux environment. It needs both computer and the CTC 100 to be on same local network(via RJ 45) connected through Ethernet cable, commands can be sent to CTC100 using Telnet which is application layer transmission protocol.

## *2.5 Assumptions and Dependencies*

A number of factors that may affect the requirements specified in the SRS include:

* Users are assumed to have a fair estimate of knowledge about the execution of CTC100 and are able to understand the automated techniques for operation of CTC100.
* CTC100 is not very robust. It crashes if commands are given in quick succession, therefore the ‘CTC100 ControlX’ maintains time difference between execution of successive controls. Hence the communication between the linux-aided system and CTC100 can be a bit slower for systems with slow processing.

# 

# 3. Specific Requirements

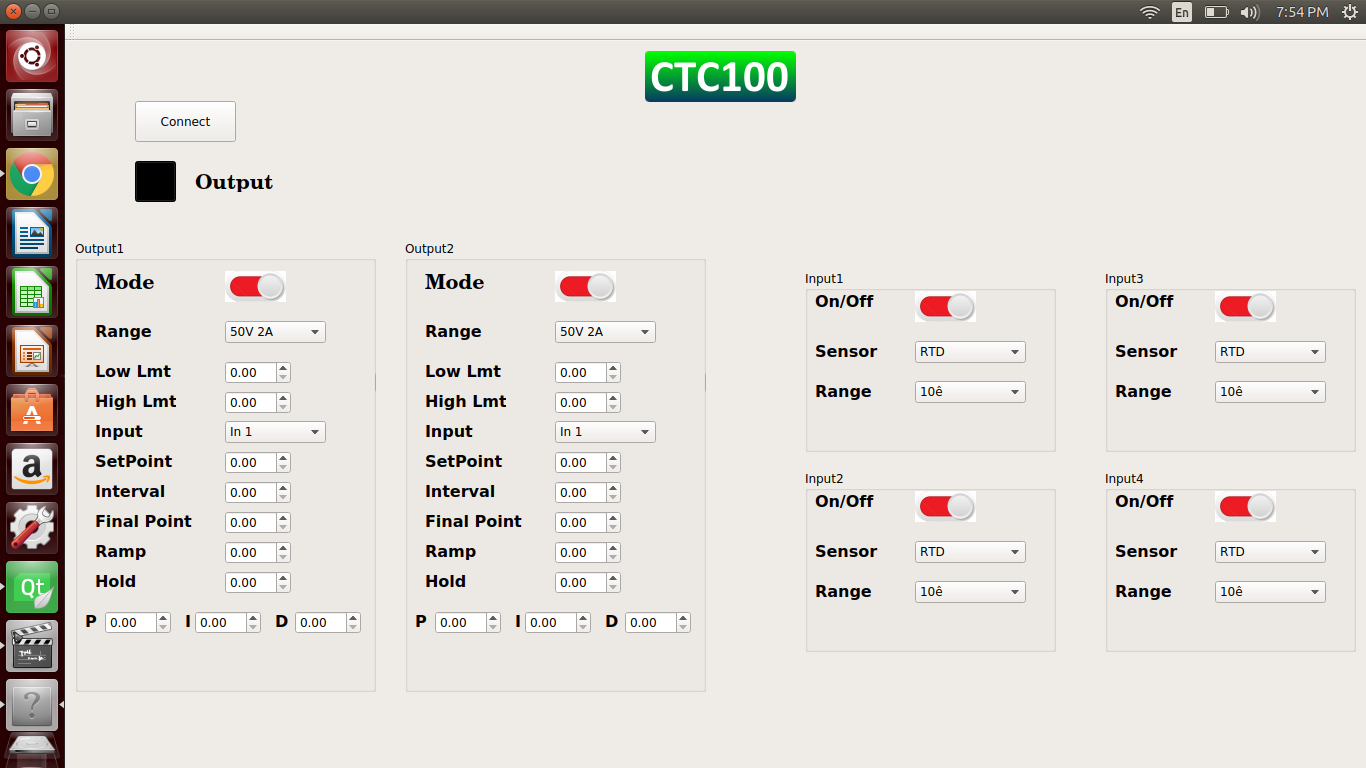
Qt software has been used to create the GUI. Commands which are to be sent to CTC100 specified by the user who inputs it using the interactive GUI are first written in a shell script, which executes commands on terminal.The Linux Terminal establishes Telnet application connection with CTC100 to send those commands to it.Telnet is an application layer protocol with underlying TCP/IP protocol for accessing remote systems.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The user interface has been designed in “Qt” a cross-platform application framework.

The user interface (prototype) looks like this-



### 

### 

### 3.1.2 Hardware Interfaces

Since the application runs via telnet application, CTC100 has to be connected to the Linux-aided system using ethernet Cross-cable.

### 3.1.3 Software Interface

1. The application ‘CTC100 ControlX’ has been designed in “Qt” which is a cross-platform application framework that is used for developing application software that can be run on various software and hardware platforms.
2. ‘CTC100 ControlX’ shall communicate with CTC100 via Telnet which is a application layer transmission protocol.

### 

### 3.1.4 Communications Interfaces

‘CTC100 ControlX’ shall communicate with CTC100 via ethernet cross-cable connection which establishes a local network between the device and system. ‘CTC100 ControlX’ uses the Telnet protocol architecture to communicate with CTC100.

## 3.2 Functional Requirements

### 3.2.1 Automating the controls of CTC100

‘CTC100 ControlX’ predominantly aims at automating most of the manual controls of CTC100.

The GUI enables the user to set various parameters like temperature setpoint, interval for change of setpoint, lower and upper limit of power output, selecting suitable sensors, setting up stable temperatures, etc.

### 3.2.2 Reading values of various parameters from CTC100

‘CTC100 ControlX’ also enables the user to monitor the changes in various scientific parameters like input temperature. It helps the user to achieve a stable setpoint within the range of tolerance specified by the user. ‘CTC100 ControlX’ has been programmed to reach successive setpoints at regular intervals of time which are again user-specified thus maximizing the user control over CTC100 and also automating the process to the fullest.

## 3.3 Non-Functional Requirements

### 3.3.1 Performance

The ‘CTC100 ControlX’ is performance optimized for a slower device like CTC100. It avoids the CTC100 crashing due to overloaded instructions by feeding inputs at stipulated intervals.

### 3.3.2 Reliability

The ‘CTC100 ControlX’ relies on Telnet application for communicating with CTC100, a much trusted network protocol.

### 3.3.3 Availability

‘CTC100 ControlX’ uses Qt, a cross-platform framework for application development to build the GUI. The communication takes place through Telnet application protocol which is also supported equally in Windows as well as Linux based operating systems.

### 3.3.4 Security

The communication between the device and system takes place through a local network established using a ethernet cross-cable. Telnet protocol aids the transmission of commands to CTC100 which is a fully secured protocol for Linux OS.

## 3.4 Design Constraints

‘CTC100 ControlX’ sends the device commands on stipulated intervals of time in order to avoid crashing of CTC100 due to overloading of instructions. This slows down the process of controlling CTC100 from the system.

## 