VISA Communication Tool Requirements Documents

Ian Absher[1], Seth Ward[1], Chenliang Wang[2], Lucien Armand Tamno[2] and Wenbo Hou[2]

[1]Tektronix, Inc.
[2]Oregon State University

November 2016

Abstract

This document aims to provide detailed information about the project, VISA Communication Tool. It will provide project overview and interface designs. Specific requirements and design constraints for the develop team also exist in this paper. At the end of this document, the develop team will specify each team member's work and the Gantt Chart to show the develop pipeline.

Contents

1 Introduction

This section briefly describes the content of this SRS document. The purpose of this document and necessary abbreviations/definitions are also in this section.

1.1 Purpose

This document aims to provide a detailed description of the requirement for the "VISA Communication Tool" (VCT) software. It will illustrate the objectives and development of the software,including the overview of the software, the design constraints, various interfaces and the baseline for any future interactions.

1.2 Scope

VCT is a Python-based VISA communication tool that manages test & measurement equipment. Users use this tool to locate devices and interact with them on the local network. For example, they can enter IP address to find a device, and send commands to check the connection between a device and the software. Additionally, the software will support users with syntax reminders that can complete incomplete commands automatically and highlight incorrect command syntax.

1.3 Definitions, acronyms, and abbreviations

Table 1: Definition

Terms	**Definition**
API	a set of subroutine definitions, protocols, and tools for building software and applications
GPIB	General Purpose Interface Bus [1]
I/O	Input/Output—The communication between an information processing system
IP address	Numerical labels of devices in a computer network that uses the Internet Protocol for communication
ISO9421	A multi-part standard for Standardization covering ergonomics of human-computer interaction. [2]
OSI	Open systems interconnection model
pyvisa-py	Python package that enables user to control all kinds of measurement device [3]
TCP/IP	The Transmission Control Protocol (TCP) and the Internet Protocol (IP), most common networking protocols
Tektronix	Test & Measurement Company
Users	Engineers who use Tektronix product
VISA	Virtual Instrument Software Architecture
VCT	Name of the software
VXI	VME eXtensions for Instrumentation. Open standard platform for automated test based upon VMEbus [4]

1.4 Reference

[1] National Instruments Corporation, "GPIB Instrument Control Tutorial," in National Instruments, 2015. [Online]. Available: http://www.ni.com/tutorial/2761/en/. Accessed: Nov. 03, 2016.

[2] International Organization for Standardization, "ISO 9241-210:2010," in ISO, 2015. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075. Accessed: Nov. 03, 2016.

[3] PyVISA, "PyVISA," in PyVISA: Control your instruments with Python.[Online]. Available: http://pyvisa.readthedocs.io/en/latest/index.html. Accessed: Nov. 07, 2016.

[4] I. Poole, "VXI Technology Tutorial," in radio-electronics.com. [Online]. Available: http://www.radio-electronics.com/info/t_and_m/vxi/vxi.php. Accessed: Nov. 1, 2016.

1.5 Overview

There are three more sections remaining in this document. The second part of this document intend to fully describe the software including the design principles, design constraints as well as the target users. While, the third chapter elaborates on specific requirements of the client in both the interface part and the functionality part. Lastly, we will end the document by providing details about the time line and milestones to be reached.

2 Overview description

This section will give an overview of the whole software. The interaction between each part and the functionality of the tool will be mentioned in this part. A summary of software features will be shown in this section. The design assumptions and constraints will be provided at the end of the section.

2.1 Product Perspective

This software allows users to interact test & measurement instruments with commands. Users can input commands to acquire and operate data from different machine. To make it convenient, the software will provide syntax supports for users.

Basically, this software implements the VISA Communication protocol to interact with measurement instruments. Instruments get orders from VCT, and then send back data to the software. All interactions happen in TCP/IP. However, we can implement USBTMC communication protocol, as long as have extra time. VCT will provide a temporary storage for each process to store the unprocessed binary data. So, users can interpret the data to debug. At the end, the software has a built-in command library that is used for the command reminder.

2.2 Product functions

VCT is a control software for test & measurement instruments with both programmatic and graphic interface. Users can use commands, built-in buttons, even keyboard shortcuts to send instructions. Instead of traditional plain text format, VCT will display collected data based on users' choices. For example, VCT can show continuous temperature change of a rod in 3D coordinates. VCT also has a subwindow to show available devices in the network, so that users can easily find device in need. Furthermore, it allows users to locate target device through IP address.

VCT can save users' working time by providing syntax reminders. When users are typing, the software will query commands based on inputting letters. Then, users will get a list of recommended command syntax as a reference. VCT automatically runs syntax check right after users type in a complete command. Incorrect commands will be highlighted.

Users are able to access to the block data of VCT to check whether the collected data is reasonable or not. Error information from a executed command will display on the command window. Users can choose to save error information and the related event into a logfile. The size of the logfile is limit to 16 KB or bigger, which depends on the further development.

2.3 Constraints

One constraint in this project is the software's compatibility. Since Tektronix has different types of product, the design team should ensure that VCT works well with different instruments. Although the VISA communication protocol is standardized, the command library could sent wrong reminders to users due to the misunderstanding. Then, the develop team needs to take care of the memory usage for each process to prevent data loss or memory leak. The last constraint is update issue. Both VISA communication protocol and Tektronix product keep changing. If VCT does not follow it, it may lead to inefficiency or even garbage data collection.

2.4 User characteristics

According to information from Textronix engineers, the develop team concludes that there are three types of users. The first type of users is engineers who use Textronix to do measurement work. Most of them are not familiar the VISA protocol, so that the design team should provide enough technical support to them including, syntax reminders and command documentations.

The second type is engineers who wrote scripts to optimize the performance of the device. Besides the syntax support, VCT allows them to access the bottom of the communication interface to debug. The last group of users are Tektronix engineers. They implement VCT to test their software on the device. With the assistance of VCT, they can access the binary data in the communication between VCT and the device, so that they can catch and fix bugs.

2.5 Assumption and dependencies

The design team assumes that all Tektronix product works well. The working computer should have enough memory for temporary memory allocation. Since VCT is VISA based, we assume that all APIs (Application programming interface) are also VISA based. The last assumption is that VCT users can understand English.

3 Specific requirements

This section contains all of the requirements of the software. It includes details about each of requirements.

3.1 User interface

The user interface of this software is user friendly and accessible. We provide a tool bar that contains basic operations. Users can click on them to find files, check connections, and etc. The main body of the user interface are the programmatic windows where users can type commands. We also provide click buttons for users to change input mode or check history.

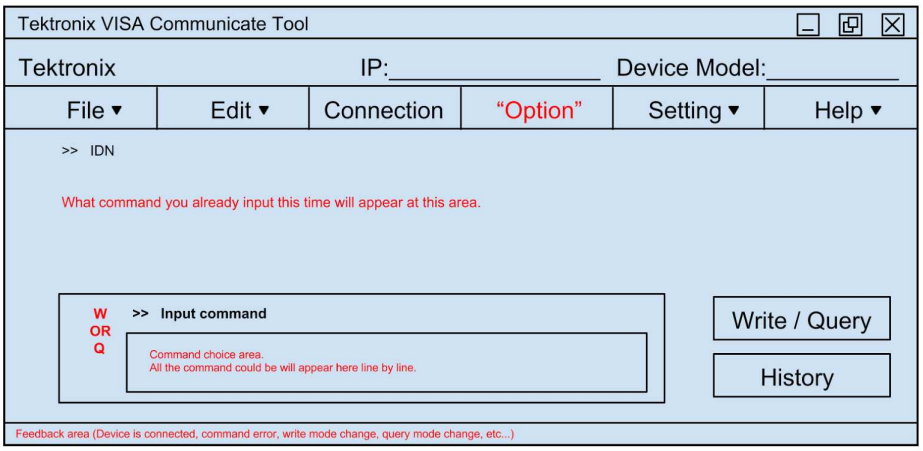


Figure 1: General UI design

As our client mentioned, we will create keyboard shortcuts for major function. For example, users can press Ctrl and S to save collected data. Also, users can customize the shortcuts. Furthermore, we will keep improving our UI design as the project goes.

3.2 Hardware interface

Since our goal is to design the management software, no formal hardware interface is needed. However, we will generate a sub window in the user interface to show status of available measurement instruments. For example, offline devices will have a red spot next to them, and on-line devices will have a green spot. Besides that, we will provide general information about each device including, the port number, IP address and etc.

3.3 Communication interface

To communicate with measurement devices, our product implements VISA communication protocol over T&Mspecific I/O interfaces such as GPIB and VXI. We will take advantage of visa32.dll that contains VISA APIs in C language to build the communication interface through TCP/IP. Also, we can use PyVisa-Py to do based on Python. We will not process data in this layer, so that users can interrupt it to debug.

3.4 Functional requirements

3.4.1 Programmatic interface

Users can input commands or use keyboard shortcuts to send instructions to devices. Besides inputting commands, users can switch to query mode where they can query commands with different key words. We will generate another input windows for users to query a command with keywords and pipe it to the command window. According to the requirements checklist, this programmatic interface needs to record 25 recently sent commands. Consequently, users can quickly re-send commands by selecting from the command history.

3.4.2 Command auto-completion

VCT will provide a list of queried commands based on letters user input. As a result, users can manually or use a keyboard shortcut to complete the command with correct syntax.

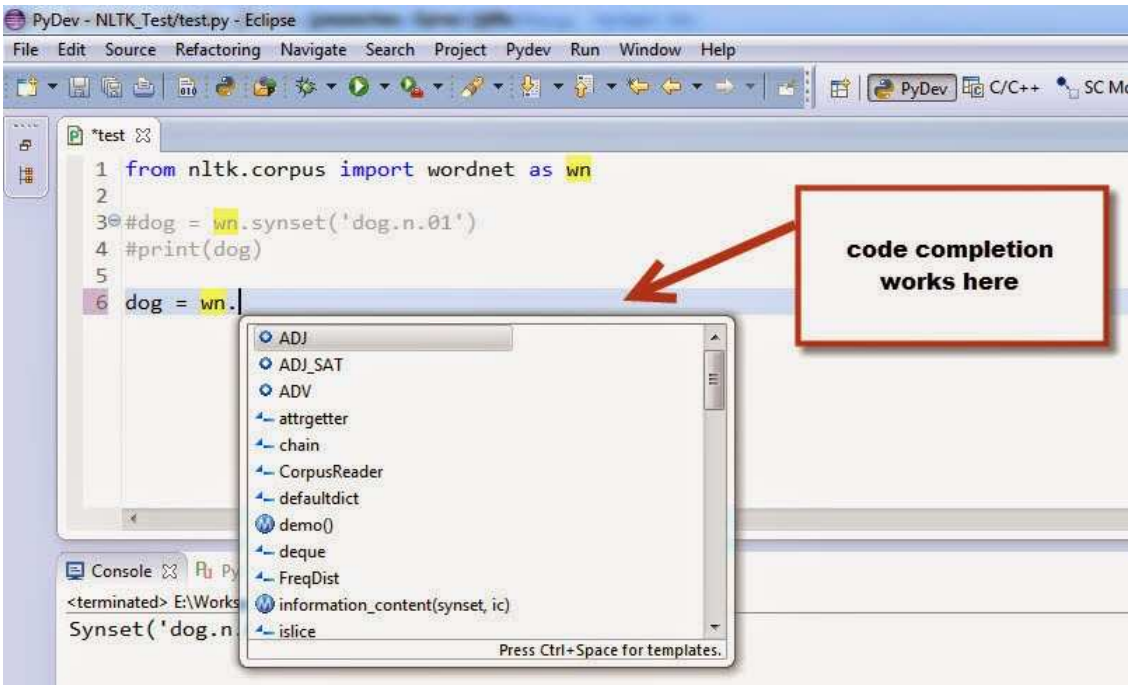


Figure 2: command auto-completion example

3.4.3 Syntax check

When the user type a command with incorrect syntax, the software will underline this command and provide a suggestion to fix it.

3.4.4 Command documentation

VCT also provide command documentations to users. The developing team will create a trigger window for user to search the command library with keywords. It will inform users about the functionality, the destination device, the received data type and etc.

3.4.5 Track device

VCT allows users to locate a device that is available in the local network. Users can do that by finding the device in the available device list or enter IP address with finding key words.

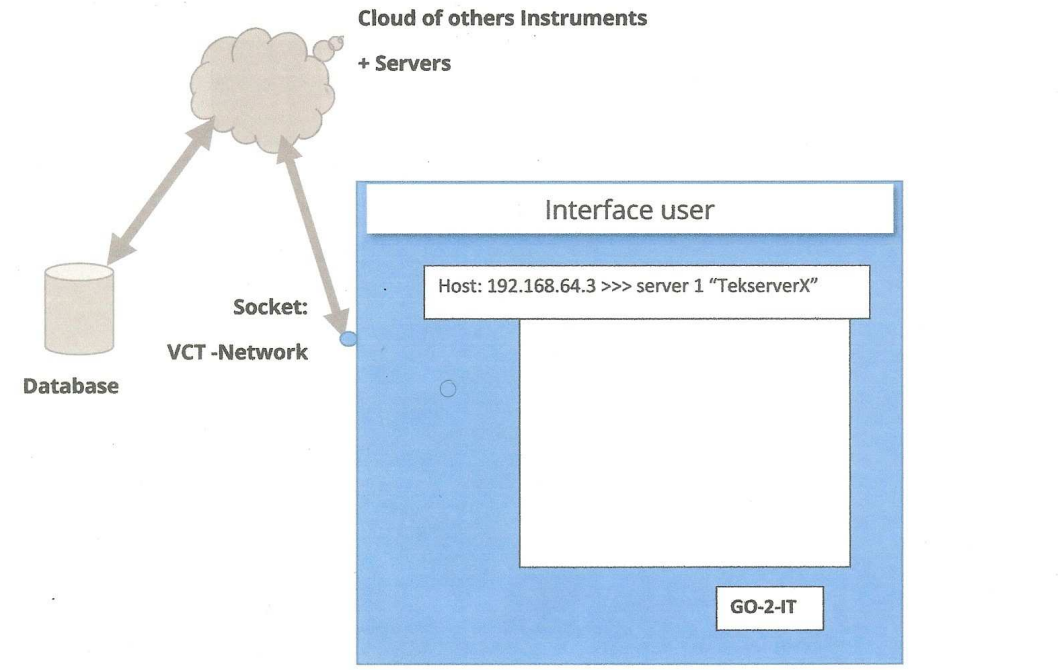


Figure 3: locate device trough IP address

3.4.6 Device Recognition

As there are different sets of commands for various devices, the VCT should differentiate between the models of devices. Once VCT connects a device, the model of device will appear in the right top corner of the interface. Furthermore, the VCT can help user to complete the commands and check syntax with the right command set.

3.4.7 History command check

In the command window, users are allowed to press up button to get a command history list. The list contains 25 histor commands can show as a floating text box over the current input line.

3.4.8 Error Reaction

VCT is able to show the error information in detail. It displays the error code with a short description like, 1073807341_Invalid access mode. Users can fix the error based on this error message.

3.4.9 Debugger

When activating debugger, VCT can capture and display packet level information. The information that VCT showed for a error includes source, destination, content, time/date, and Length. Debugger will also provides access

to block data to check the communication between the software and the device.

3.5 Software system attribute

3.5.1 Reliability

To be reliable, the software has to established the end to end communication. First, the device to pull data from should be seen on the GUI as on (preferably green color icone). second, with the assumption that others systems in between work just fine(Operating System, network if needed) the software should be able to accurately read from and/or write to any device connected (oscilloscope, but also server if implementing Database) and all this according to functionality of auto-completion of commands and if the functionality is not supported by the connected device, a message is triggered and displayed on the GUI, inviting the user to use the document of commands provided or other assistance. However, if the user enters the wrong word, the list is shrunk down to all the existing commands possible and the word itself is highlighted to mark the wrong spelling. And also the software will accurately and securely commit and retrieve data wherever they have been stored and its deliveries in compliance with the format expected.

3.5.2 Availability

Being a load and run software, VCT has to operate on any Operating System we can find on the market today (Windows, MacOs, Linux, and the oncoming Fuchsia).the software should be available in the cloud, the cloud being the various networked storage places determined by the client and easily accessible by a user to download in reasonable period of time.

3.5.3 Security

As far as the security aspect is concerned, a user working on the VCT software is forbidden to pull out data he/she is not intended to. To mean that, anyone working on any attached instrument except authorized, will able to access others informations out the scope of the work to be done. Also, the VCT might implement the licence agreement in order to make sure that any user accessing it has the right to do so. In addition, the software should not contain spaghetti logic Module in its structure. To say that, no implemented module has to overlap with others.

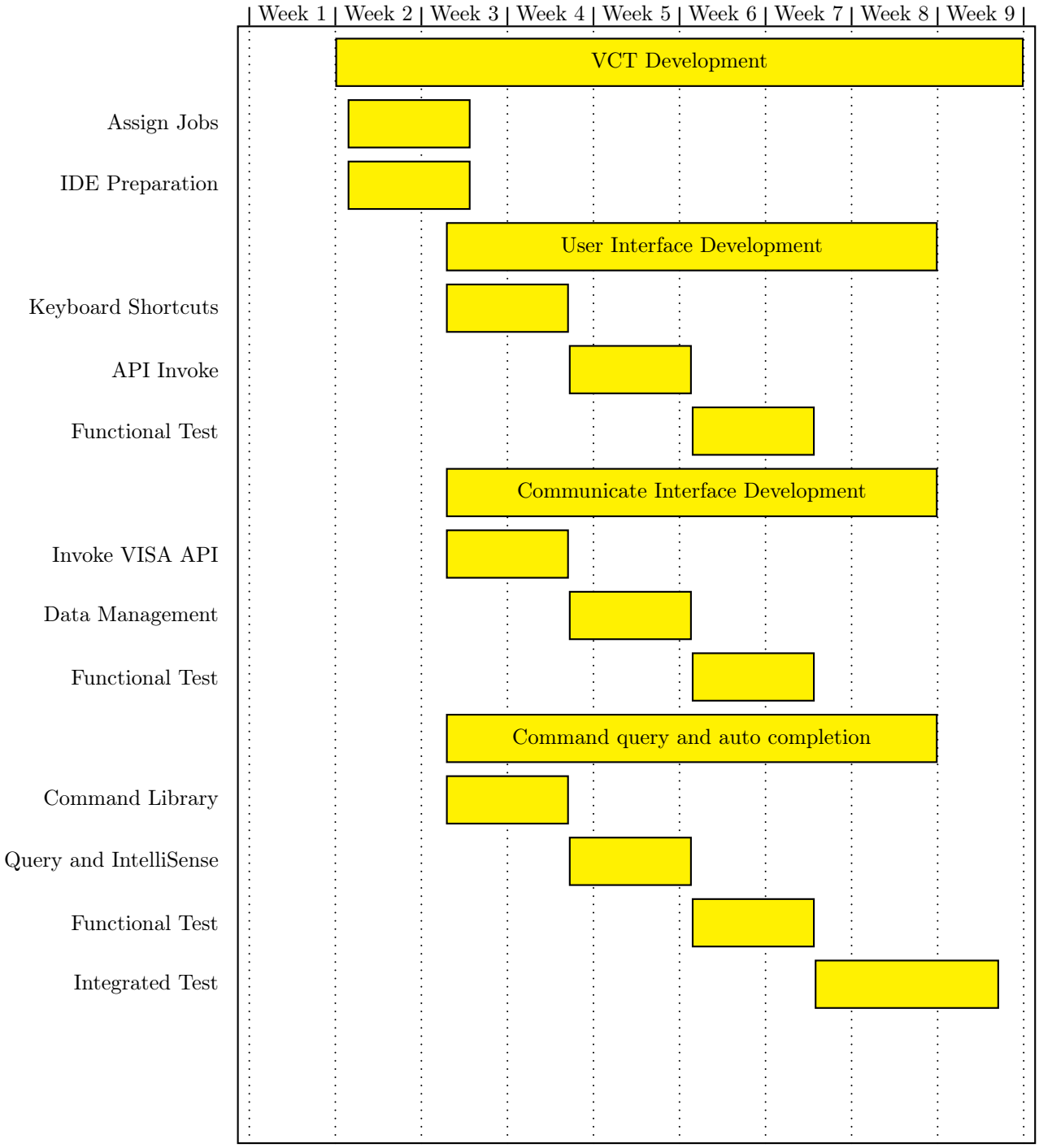
3.5.4 Maintainability

The maintainability being defined as the capability of the software to undergo some rearrangements of it code in order to yield better performance and even to integrate new modules and thus, to be easily updatable. Nonetheless, the software can allow a user to perform some light maintenance on the user interface.

3.5.5 Portability

Here comes into play, one of the key feature of VCT which is load and run, the means that the user on any operating system would be able to connect to the server on the accessible network download the application and runs it on the platform of his or her choice. In other words, the software has to guarantee its high capability of portability without any crash, miscommunication over the communication interface which, in fact, is transactional point where the VCT hands over the communication to the operating system by if possible encrypting raw data (at application layer of OSI model; OSI: open systems interconnection model).

4 Gantt Chart



Please Sign in this page: