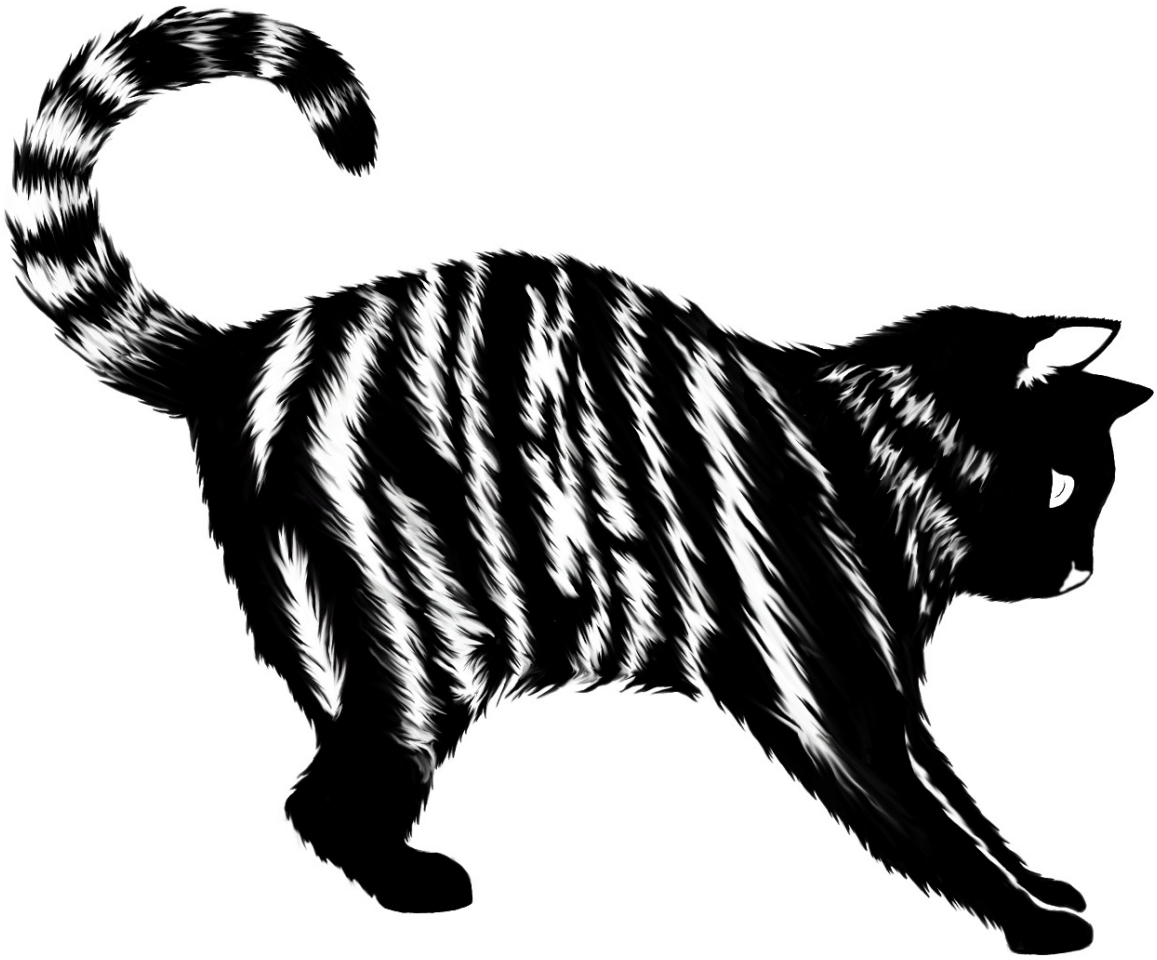

System Version Description

for the

Garden Control System

DCN: KNEADSVD20231222-P1:42

Revision Date: 29 Apr, 2024

Prepared by:

Zachary Steinberg
Garden Control System
Washington, D.C. 20001 USA

DOCUMENT CHANGE HISTORY

The following table is a simple list of released revisions sent for review. Records of reviews and the review artifacts are saved with reviewer information in the The KNEAD Projectartifact repository.

Change Record

Date	Version	Author(s)	Change Reference
22 Dec 2023	P1	Lewis Collier	Preliminary DRAFT version

Each subsequent "section" outlines changes in each release.

Items in this version that are marked with change bars have been modified from the most recent previous version (e.g. P3 changes from P2) or are new as of the current revision. A list of all changed items may be found in the Index section under the heading "All Changes This Version".

Draft P1 Preliminary version of this document.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

Scope

This document provides the System Version Description (SVD) for the Garden Control System. The system will be referred to as the GCS.

This document is generally cited as [1].

1.1 Identification

The Garden Control System described in this document shall be known as GCS version 1.0. The Garden Control System may also be referred toas GCS v1. The SVD and other design documents are intended for those interested in replicating or building Garden Control System.

1.2 System Overview

The Garden Control System will be able to measure moisture levels and control irrigation in raised garden beds. GCS is being developed by Zachary Steinberg and sponsored by University of Maryland Graduate Engineering. The operator and maintaner of GCS will also be Zachary Steinberg. The GCS will be operated outside along raised garden beds. GCS is designed to be used by home gardeners. It is not intended for industry. GCS will be controlled by a Raspberry Pi Pico W microcontroller board.

Figure 1 shows the development kit used for the GCS system. This is an image of the

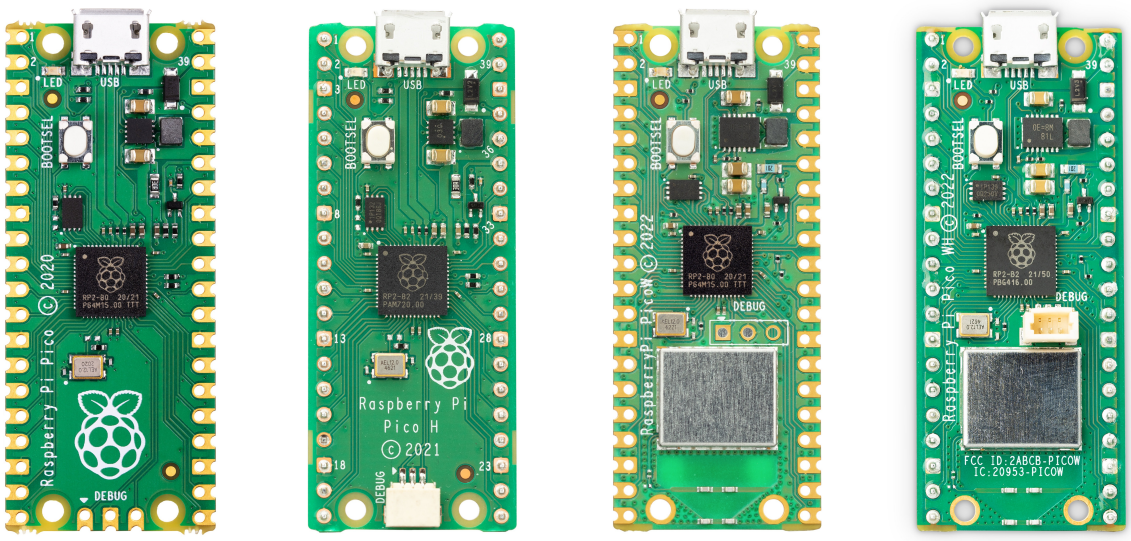

Figure 1: Raspberry Pi Pico W microcontroller board

Raspberry Pi Pico W microcontroller board.

1.3 Document Overview

This section provides information about this document's security/privacy considerations, contents, structure, and version information. As of now, there are no security or privacy concerns. This will be updated if applicable.

1.3.1 Contents and Structure

This document format is based upon the guidance in the SDP DID [2], but is tailored to reflect current quality management system (QMS) requirements for many of the items described in the DID. The system development plan information also follows the guidelines of ISO-12207 [3] and MIL-STD-498 [4] (from which ISO-12207 originated). It is expected that significant coverage of the SDP DID items will simply be references to external QMS or CMMI documentation. Thus, agencies using this artifact will formalize it to serve as a checklist for tailoring each effort per their own processes and procedures.

This tailoring also includes planning for hardware aspects as may be needed for the complete system development efforts. The SDP DID mainly covered integration of software onto hardware. This artifact includes sections that cover the development of hardware as well as development of the software. Integration and testing are also included. But, as noted above, all apsects of development are expected to illustrate the existing QMS steps to be followed based on organizational processes and the project's operational needs.

This document follows a tailored SVD sub-section order of:

Section 1 provides an overview of the system and this document.

Section 2 lists general and application-specific reference documents as well as glossary terms and acronyms.

Section 3 summarizes the required work, with references to existing QMS processes and procedures as applicable:

- program status / acquisition strategy,
- SDLC Situation,
- plans for development of requirement artifacts,
- plans for overall documentation development,
- schedule / resource constraints, and
- other requirements/constraints.

Section 4 Plans for system development

- Hardware development processes and plans,
- Firmware development processes and plans,

- Software development processes and plans,
- Integration plans,
- Testing plans, and
- Other development activities per organizational processes.

Section 5 Plans for system transistion

- Configuration Management plans,
- Release plans,
- User Support plans, and
- Other system transition activities per organizational processes.

Section 6 Management and control activities

- SETR events (internal previews and with customer),
- Skills and resources needed,
- Schedule Development and Monitoring, and
- Other management and control activities per organizational processes.

Section 7 if needed, lists any general notes as may be applicable.

Appendices if needed, provide additional information as may be needed.

If this text is visible, the first instance of each section may display a summary of data item description (DID) information shown in this font. These are displayed in small capital font and are not part of the formal document.

1.3.2 Document Version Information

This document was produced in LaTeX and *BibLaTeX/Biber*. The editing and document preparation were performed using MiKTeX version 2.9 with the build option [LaTeX ⇒ PS ⇒ PDF]. The LaTeX*svn-multi* package was used to glean SVN tracking information, when files are stored in an "SVN" version control system. The style `KNEADdocument` was used to provide the LaTeX and *BibLaTeX/Biber* formatting details.

This revision of this document has the following properties:

Tracking Item	Data
Repository	https://svn.riouxsvn.com/kneadlatxinputs/ExampleArtifactFolders/7%20-%20SVD/KNEAD_SVD.tex
Author	LCollier
Revision	595
Rev Date	2017-05-30 14:07:20Z
Print Date	29 Apr, 2024 01:31
KNEADdocument Version	1.00
KNEADdocument Date	2021/12/05

CHAPTER 2

References

This section provides a list of referenced items for this document.

2.1 Acronyms and Abbreviations

This section defines acronyms and abbreviations used in this and related documents.

Table 1: Acronym Definitions

Acronym	Definition
GCS	Garden Control System
UMD	University of Maryland
MAGE	Maryland Applied Graduate Engineering
ENPM	Engineering Professional Masters
End of acronym definition table	

2.2 Glossary and Definitions

This section defines glossary terms used in this and related documents.

Table 2: Glossary Terms and Definitions

Glossary Term	Definition
Communications	Communication is information transfer, among users or processes, according to agreed conventions.
Customer	The local government project lead who is acting as a general manager for the sponsor to ensure that the contractor team executes the project according to stakeholder goals.
End of glossary terms table	

2.3 Referenced Documents

This section lists the referenced documents for this document. The references are categorized into two categories:

External Documents not directly associated with this project.

Project Documents that are directly associated with this project.

2.3.1 External Documents

[2] DI-IPSC-81442A. *Data Item Description for System Version Description*. Jan. 11, 2000.

[3] IEEE and EIA. *Software life cycle processes*. Mar. 1998.

[4] MIL-STD-498. *Military Standard Software Development and Documentation*. Dec. 31, 1994.

[5] Raspberry Pi Ltd. *Raspberry Pi Pico C/C++ SDK*. Feb. 17, 2024. URL: https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-c-sdk.pdf.

[6] Raspberry Pi Ltd. *Raspberry Pi Pico C/C++ SDK*. Feb. 17, 2024. URL: https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf.

2.3.2 Project Specific Documents

[1] The KNEAD Project. *System Version Description for the KNEAD Example Project*. Dec. 31, 2023.

CHAPTER 3

Version Description

This chapter will provide an inventory of all used software and hardware, provide change information for the Garden Control System, and provide installation instructions for required software.

3.1 Inventory of Materials

The materials used for the Garden Control System are:

- Raspberry Pi Pico W
  - Raspberry Pi Pico W Datasheet

The Raspberry Pi Pico W is a RP2040-based microcontroller board with wireless capabilities.

3.2 Inventory of Software

The software used for the Garden Control System are:

- pico-sdk: Release SDK 1.5.1 June 13, 2023
  - pico-sdk Github repository
- pico-examples: Release Tag sdk-1.5.1 June 11, 2023
  - pico-examples Github repository

Both the pico-sdk and pico-examples repositories are provided by the Raspberry Pi Ltd organization. The pico-sdk and pico-examples repositories work on any Raspberry Pi Pico board.

The pico-sdk SDK is a C/C++ SDK designed for the Raspberry Pi Pico family of microcontroller boards. The documentation for pico-sdk can be found in the SDK documentation [5]. The pico-examples repository contains sample code and projects that are provided by the Raspberry Pi Ltd organization. The sample projects use Release 1.5.1 of the pico-sdk.

3.3 Adaptation Data

There is currently no unique-to-site data for the Garden Control System. Updates will be made to this section when applicable.

3.4 Related (Third Party) Documents

The following is a list of third party documents associated with the software used for the Garden Control System:

- [6] Raspberry Pi Pico Getting Started Guide
- [5] Raspberry Pi Pico C/C++ SDK Official Documentation

3.5 Installation Instructions

This section will provide an overview for how to install the pico-sdk and pico-examples repositories.

3.5.1 Installing pico-sdk

For detailed steps on how to install the pico-sdk, please refer to the [6] Raspberry Pi Pico Getting Started Guide. The pico-sdk can be found in the github repo: `github.com/raspberrypi/pico-sdk` The following will list the necessary steps to install the pico-sdk:

1. Install CMake (at least version 3.13), and GCC cross compiler
2. Clone the Raspberry Pi Pico SDK repository using git clone
3. Set the ENV variable PICO_SDK_PATH to the SDK location in your environment.
4. Setup a CMakeLists.txt file

These steps will install and configure the pico-sdk repository onto your system.

3.5.2 Installing pico-examples

For detailed steps on how to install the pico-examples, please refer to the [6] Raspberry Pi Pico Getting Started Guide. pico-examples can be found in the github repo: `github.com/raspberrypi/pico-examples` The following will list the necessary steps to install the pico-examples:

1. Clone the Raspberry Pi Pico Examples repository using git clone

3.5.3 Quick Pico Setup via Install Script

The Raspberry Pi organization provides a setup shell script to configure your system with the pico-sdk and pico-examples script. The Getting Started With Pico [6] guide provides more information about the setup script. The following are the necessary steps to run the install script:

1. Get the script via terminal command: wget `https://raw.githubusercontent.com/raspberrypi/pico-setup/master/pico_setup.sh`
2. Make the script executable with: chmod +x pico_setup.sh
3. Run the script with: ./pico_setup.sh

3.6 Possible Problems and Known Errors

There are no possible problems or known errors at this time. This section will be updated if and when applicable.