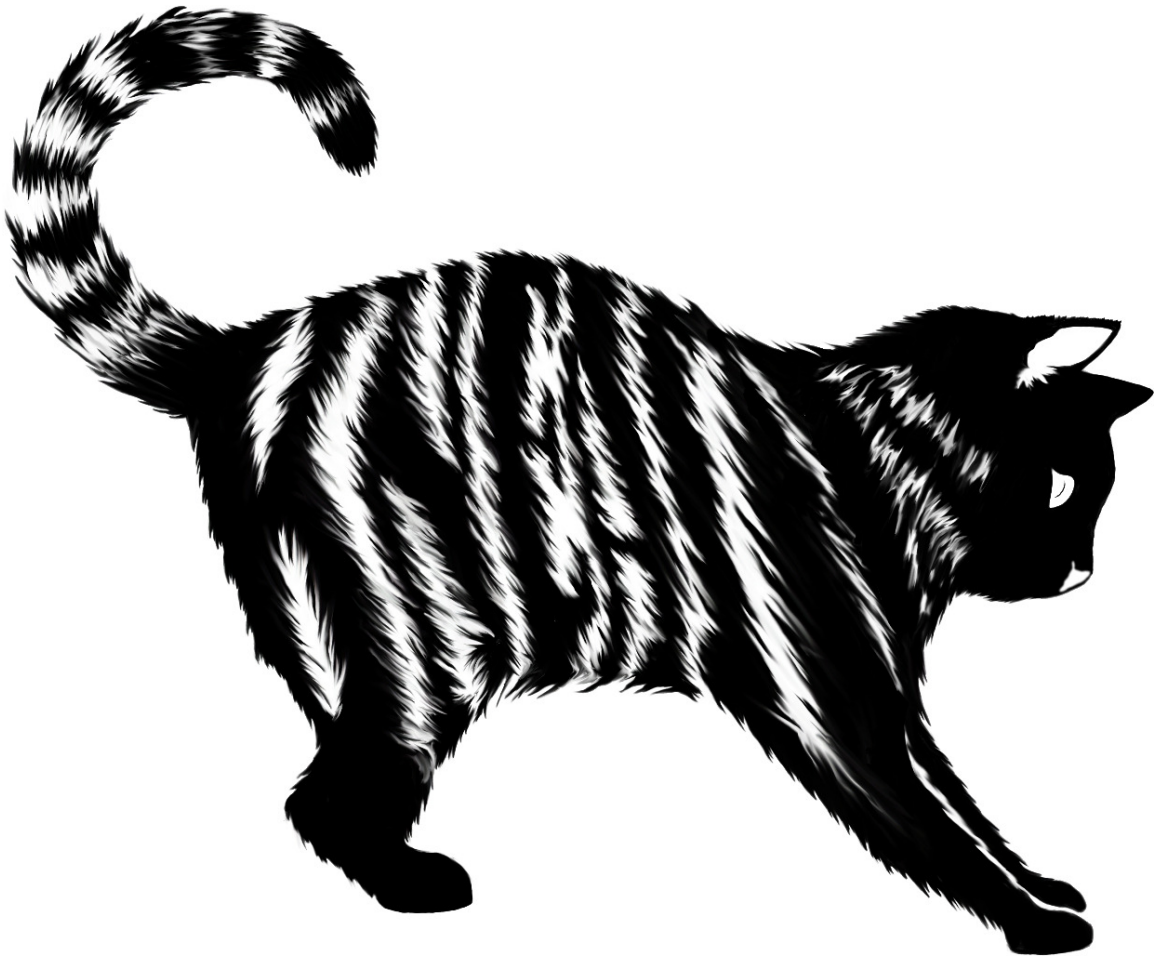

System Development Plan

for the

Garden Control System

DCN: KNEADSDP20231224-P1:44

Revision Date: 27 Apr, 2024

Prepared by:

Lewis Collier
The KNEAD Project
Lexington Park, MD 20653 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
17 Feb 2024	P1	Zachary Steinberg	1st 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

CHAPTER 1

Scope

This document provides the System Development Plan (SDP) for the Garden Control System. The system will be referred to as the GCS.

1.1 Identification

The Garden Control System described in this document shall be known as GCS version 1.0. Garden Control System is the unofficial name for GCS. An official name will be determined at a later time. GCS is currently in development. No versions or releases are available at this time.

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 high-level architecture for the GCS system. This is an image of the

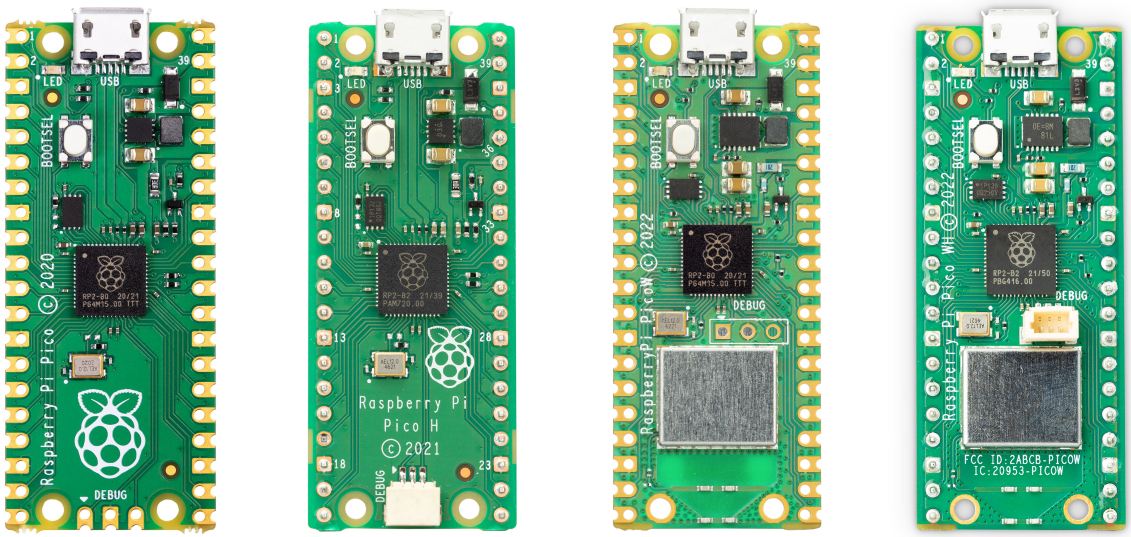

Figure 1: Raspberry Pi Pico W microcontroller board

Raspberry Pi Pico W microcontroller board.

The general concept of operations (CONOP) for this system is ...TBD....

1.3 Document Overview

This section provides information about this document's contents, structure, and version information.

1.3.1 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/1%20-%20SDP/KNEAD_SDP.tex`
Author	LCollier
Revision	595
Rev Date	2017-05-30 14:07:20Z
Print Date	27 Apr, 2024 17:58
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

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

[15] 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. *Operational Concept Description for the KNEAD Example Project*. Dec. 31, 2023.

[2] The KNEAD Project. *System Development Plan for the KNEAD Example Project*. Dec. 31, 2023.

[3] The KNEAD Project. *System Performance Specification for the KNEAD Example Project*. Dec. 31, 2023.

[4] The KNEAD Project. *System Subsystem Specification for the KNEAD Example Project*. Dec. 31, 2023.

[5] The KNEAD Project. *System User Manual for the KNEAD Example Project*. Dec. 31, 2023.

[6] The KNEAD Project. *Hardware Requirements Specification for the KNEAD Example Project*. Dec. 31, 2023.

[7] The KNEAD Project. *Software Requirements Specification for the KNEAD Example Project*. Dec. 31, 2023.

[8] The KNEAD Project. *Interface Requirements Specification for the KNEAD Example Project*. Dec. 31, 2023.

[9] The KNEAD Project. *System-Subsystem Design Description for the KNEAD Example Project*. Dec. 31, 2023.

[10] The KNEAD Project. *System Test Plan for the KNEAD Example Project*. Dec. 31, 2023.

[11] The KNEAD Project. *System Test Specification for the KNEAD Example Project*. Dec. 31, 2023.

[12] The KNEAD Project. *System Test Report for the KNEAD Example Project*. Dec. 31, 2023.

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

CHAPTER 3

Required Work Overview

This chapter will provide and overview of the required work, project constraints, project status, and project dependencies for the Garden Control System.

3.1 Program Status

The Garden Control System program exists within the UMD MAGE class ENPM818I: Variable Topics in Engineering: Embedded Software Design and Optimization under the supervision of Dr. W. Lewis Collier.

3.2 SDLC Situation

This is an original project. All materials for this project are generated for the purpose of designing the Garden Control System. Design materials will be completed by the end of ENPM818I.

3.3 Requirement Plans

The requirement plan for the Garden Control System project is to design and document the design plan for the Garden Control System.

3.4 Documentation Plans

The documents listed

The following documents are listed here just to test reference generation. A "real" SDP would reference these as applicable for the project.

- The KNEAD Project, *Operational Concept Description for the KNEAD Example Project* [1] is the OCD, which outlines the project overall so, generally, it is created first.
- The KNEAD Project, *System Development Plan for the KNEAD Example Project* [2] is this document.
- The KNEAD Project, *System Performance Specification for the KNEAD Example Project* [3] is the SPS, which should come from the customer or end user, but often is generated by the developer with customer approval.
- The KNEAD Project, *System Subsystem Specification for the KNEAD Example Project* [4] is the SSS that is the developer's design specification to meet the SPS requirements.
- The KNEAD Project, *System User Manual for the KNEAD Example Project* [5] is the SUM that acts somewhat like part of the SSS since it illustrates the UI design part of the SSS, but in a separate artifact that also can be used as a standalone users' manual.

- The KNEAD Project, *Hardware Requirements Specification for the KNEAD Example Project* [6] is a HRS, which often is not used for smaller projects but can have multiple instances for large projects to more fully detail hardware design.

- The KNEAD Project, *Software Requirements Specification for the KNEAD Example Project* [7] is a SRS, which often is not used for smaller projects but can have multiple instances for large projects to more fully detail software or firmware design.

- The KNEAD Project, *Interface Requirements Specification for the KNEAD Example Project* [8] is the IRS, which often is not use but may be needed, even if HRS or SRS artifacts are not, to fully document detailed interfaces such as Application Programming Interfaces (API) or other detailed mechanical or electrical interfaces.

- The KNEAD Project, *System-Subsystem Design Description for the KNEAD Example Project* [9] is the SSDD that provides a road map to the design and other design details needed to understand the hardware and software design.

- The KNEAD Project, *System Test Plan for the KNEAD Example Project* [10] is the STP that highlights the planning for system testing.

- The KNEAD Project, *System Test Specification for the KNEAD Example Project* [11] is the STS, which is sometimes called a test procedure. There could be multiple of these based on the overall project size.

- The KNEAD Project, *System Test Report for the KNEAD Example Project* [12] is an STR that documents the results of a given test. Multiple instances are expected based on the test plan. And, there could be multiple versions of a given test plan to document repeated occurrences of a given test specification/procedure.

- The KNEAD Project, *System Version Description for the KNEAD Example Project* [13] is an SVD that documents a given release of a system. Multiple versions of these "release notes" are expected, with one SVD issued for each system release cycle.

3.5 Schedule and Resource Constraints

The ENPM course and Garden Control System project involves five development milestones and five design milestones. The development assignments must demonstrate understanding of the stated embedded development objectives. The design assignments require draft completions of the listed artifacts and that will be the basis of the design artifacts for

the final project. The following is the order in which assignments must be completed for this course and project.

- Development Assignment 1: Dev Kit and Tools
- Design Artifact 1: SDP and SVD
- Development Assignment 2: Debugger and Emulator
- Design Artifact 2: OCD and SPS
- Development Assignment 3: IO Streams and Interrupts
- Design Artifact 3: SSS
- Development Assignment 4: Security Measures
- Design Artifact 4: SSDD
- Development Assignment 5: Networking
- Design Artifact 5: STP and STS

There are currently no known resource constraints. This project is only designed by one person. Design work is the culmination of my work throughout the class.

3.6 Other Constraints

Current constraints for the Garden Control System project are unknown. As design and development of the Garden Control System project progresses, this section will be updated.

CHAPTER 4

System Development Plans

This chapter will provide a brief overview of the required hardware and firmware for Garden Control System.

4.1 Hardware Development Plans

There will be no hardware development for this project. All development will take place on a Raspberry Pi Pico W development kit.

4.2 Firmware Development Plans

The firmware development plan will follow the development assignment outline for the course ENPM818I. The development assignments are as follows:

- Dev Assignment 1
  - Development kit and tools are operational
- Dev Assignment 2
  - Debugger and Emulator tools for the devlopment kit are operational
- Dev Assignment 3
  - Demonstrate necessary IO Stream and Interrupt Handler for the project.
- Dev Assignment 4
  - Demonstrate security measures necessary for the project.
- Dev Assignment 5
  - Implement networking capabilities for the project.

4.3 Software Development Plans

4.3.1 Raspberry Pi Pico C/C++ SDK Overview + Installation

The main documentation for the Raspberry Pi Pico C/C++ SDK can be found in Raspberry Pi Ltd, *Raspberry Pi Pico C/C++ SDK*, `https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-c-sdk.pdf` [14]. The document provides information about the SDK architecture, Programmable I/O, and Library documentation.

Installation instructions can be found in Raspberry Pi Ltd, *Raspberry Pi Pico C/C++ SDK*, [15]. Chapter 1 of the Getting Started guide provides instructions for how to setup the Raspberry Pi Pico using the pico_setup script. Chapter 2 of the Getting Started guide provides instructions for how to build, install, and work with the C/C++ SDK.

4.3.1.1 Setting Up the Raspberry Pi Pico

To setup a Raspberry Pi Pico, follow chapter 1 of Raspberry Pi Ltd, *Raspberry Pi Pico C/C++ SDK*, [15]. Use the pico_setup.sh installation script. This installation script will:

- Create a directory called pico
- Install required dependencies
- Download the pico-sdk, pico-examples, pico-extras, and pico-playground repositories
- Define ENV variable paths in the .bashrc file.
- Build the blink and hello_world examples in pico-examples/build/blink and pico-examples/build/hello
- Download and build picotool, and copy it to /usr/local/bin.
- Download and build picoprobe.
- Download and install Visual Studio Code
- Download and compile OpenOCD
- Install the required Visual Studio Code extensions
- Configure the Raspberry Pi UART for use with Raspberry Pi Pico

4.3.1.2 Installing the Raspberry Pi Pico C/C++ SDK

To install the Raspberry Pi Pico C/C++ SDK, follow chapter 2 of Raspberry Pi Ltd, *Raspberry Pi Pico C/C++ SDK*, [15]. Section 2.1 instructs to install the SDK from the pico-sdk github repository. It also links to the pico-examples github repository that provides a set of example applications that are written using the C/C++ SDK. Additionally, in sections 2.2 and 2.3, there are installation instructions for required tools to build and run applications in pico-examples and information for how to update the SDK. The tools include:

- gcc-arm-none-eabi
- libnewlib-arm-none-eabi

- build-essential

The software development plan for this project is still ...TBD.... This section will be updated in the future.

4.4 Integration Plans

The integration plans for this project are still ...TBD.... This section will be updated in the future.

4.5 Testing Plans

All firmware and software for Garden Control System will be accompanied with unit tests. An open source tool will be selected to audit the code base and there will be a set threshold for code coverage. Otherwise, the formal testing plan for Garden Control System is still ...TBD... and will be updated in the future.

4.6 Other Development Activities

Other development activities will be added to this document in the future. As of now, this section is ...TBD....

CHAPTER 5

System Transition Plans

The following chapter details the system transition plans for Garden Control System.

5.1 Configuration Management Plans

All code repositories for Garden Control System will be housed in Github. All code repositories will be controlled by Git.

New code will be added and integrated into the Garden Control System code base on a feature by feature basis. Pull Requests will be subject to a specified template, minimum code test coverage threshold met, and passing of other specified tests referenced in section 4.5. Any PRs that fail to meet the template specifications and testing requirements will be rejected until the requirements are met. A PR template, code coverage threshold, and required tests will be detailed in section 4.5.

5.2 Release Plans

Release plans for Garden Control System are still ...TBD.... This section will be updated in the future.

5.3 User Support Plans

User Support Plans for Garden Control System are still ...TBD.... This section will be updated in the future.

5.4 Other Transition Plans

Other transition plans for Garden Control System are still ...TBD.... This section will be updated in the future.

CHAPTER 6

Management and Control Activities

This chapter will cover the technical reviews, skills and resources, and scheduled development and monitoring activitis for the Garden Control System project.

6.1 Technical Review Events

The technical review events for the Garden Control System project will be the design assignments throughout ENPM818I.

The design assignments are as follows:

- Design Assignment 1
  - Create drafts for the SDP and SVD artifacts.
- Design Assignment 2
  - Create drafts for the OCD and SPS artifacts.
- Design Assignment 3
  - Create drafts for the SSS artifact.
- Design Assignment 4
  - Create drafts for the SSDD artifact.
- Design Assignment 5
  - Create drafts for the STP and STS artifacts.

The artifacts for each design assignment are expected to include pertinent information for the Garden Control System project and specifics to the corresponding development assignment it's paired with.

6.2 Skills and Resources Needed

As of now, it is unknown what skills and resources will be needed for the Garden Control System project. This section is still ...TBD... and will be updated when more information is available.

6.3 Scheduled Development and Monitoring

The schedule development and monitoring for the Garden Control System project is currently unknown. This section is still ...TBD... and will be updated when more information is available.

6.4 Other Management and Control Activities

Other management and control activities for the Garden Control System project is currently unknown. This section is still ...TBD... and will be updated when more information is available.

CHAPTER 7

Notes

ALL-Notes :: This section shall contain any general information that aids in understanding this document (e.g., background information, rationale, etc.)

There are currently no notes for the Garden Control System project. When and if notes are added, they will be updated here.

APPENDIX

Other Info

This section provides other information, as necessary, to document the system development plan.