**Software Architecture Specification**

**Small Volume Ink System**

**Contents**

1. **Introduction**
2. **System Context**

2.1 System Boundaries

2.2 Users

2.3 External Systems

1. **Architecture Overview**

3.1 Information Flow

3.2 Producer/Consumer

3.3 State Machine

3.4 Program Routines

1. **Functional Architecture**

4.1 Components

4.2 User Input System

4.3 Master Controller

4.4 Logic Control/Interface

4.4.1 Pressure

4.4.2 Temperature

4.5 Monitoring System

4.6 DAQ Signals

4.7 Optional Blocks

1. **Architecture Design Decisions**

**1. Introduction**

Domino Printing, founded in 1978, is a well-established company focusing on printing technologies. It is part of the Domino Group, a large worldwide network pushing industry standards with outstanding quality. Domino is involved in every aspect of its systems, from conceptualization and research to marketing and sales.

A division within Domino, Marking Materials, spends a considerable amount of time developing new inks based on client requirements. As with most development processes, testing plays an important role to ensure reliability and satisfaction of requirements. Innovative ways to increase efficiency of any established testing routines/practices are highly valued within this division.

The Small Volume Ink System is conceptualized to address the need for a more efficient and analyzable ink delivery system. The project is tailored to be completed over a 3 month period, with considerations to further development/implementation.

**2. System Context**

**2.1 System Boundaries**

There are many different factors that exert an influence on the software system. The system boundaries are outlined in below to give a high level model.

Figure 1 – Context diagram of software system

C:\Users\v_wong\Downloads\System Context Diagram ff.png**2.2** **Users**

The software system will involve two main roles associated with the users, that of an operator and that of an analyst.

The operator interacts with the system as part of a larger system to complete a mechanical task.

“*As an operator, I want to be able to provide continuous ink delivery to the printhead with a small volume so that I do not spend resources preparing large volumes of fluid.”*

The analyst interacts with the system with the small volume ink system to ascertain information regarding the fluid being used.

*“As an analyst, I want to be able to obtain diagnostic metrics from the system so that I can further development with the information.”*

**2.3 External Systems**

The user interface will be centralized on a PC/Laptop running the LabVIEW program. User commands will be interpreted as a series of keystrokes or mouse clicks.

The software system will also interface with a data acquisition device (DAQ) as a means of communicating with hardware circuits for inputs and outputs.

Reference to hardware specification document?

**3. Architecture Overview**

**3.1 Information Flow**

**3.2 Producer/Consumer**

The overarching design for this architecture is that of a producer utilizing multiple, parallel consumers. This allows user commands to be put into a queue while previous operations are being processed. The disjointed nature between work orders and processes can be taken advantage of, allowing individual timing considerations and simplifying integration of extra functions.

**3.3 State Machine**

**3.4 Program Routines**

These will be either static protocols running predetermined processes, such as initialization, shutdown or in the context of ink systems – a flush function, or dynamic looping processes such as PID control of pressure/temperature.

More routines?

**3.5 flowchart diagram + more**

**4. Functional Architecture**

Software broken into components/modules

-detailed explanation of components and interactions with rest of system

State machine diagram

**4.1 Components**

The software architecture of this system is based heavily around modular design. The system is broken into several components, each with a distinct function.

**4.2 User Input System**

Data flow of this software system, outside of looping states, originates from user input. User input events will trigger certain routines to be executed. The software will process the input and any associated variables/parameters to determine the desired state of multiple hardware components.

Controls for the user interface will be represented in mainly two forms: Boolean switch buttons and analogue dials/open fields for parameter entry.

Display elements of the user interface will also be present to provide feedback information on the system. This will allow monitoring of the status of the system, as well as detailed information for developmental analysis.

**4.3 Master Controller**

**4.4 Logic Control/Interface**

**4.4.1 Pressure**

**4.4.2 Temperature**

**4.5 Monitoring System**

**4.6 DAQ Signals**

This information will then be generated into waveform signals to be directed towards the data acquisition module. Two way communication with the DAQ is essential to the functionality of this system, hence signals from the data acquisition module are also required.

AO, DO etc

**4.7 Optional Blocks**

**C:\Users\v_wong\Downloads\Block Diagram 2 (1).png**