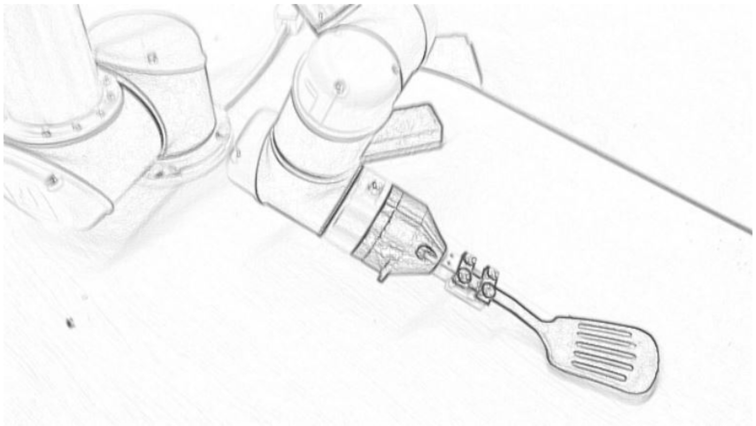
Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2018



Team Tassium
Master Chef

Anthony Tatowicz
Jesse Daniel Mitchell
Todd Brewer
Linh Vu

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	5.01.2018	TB	Initial creation
0.2	5.11.2016	TB	Final Draft

Contents

List of Figures

List of Tables

1 Introduction

The UR5 Master Chef is a product that aims to alleviate the need for manual labor in a cooking environment. The purpose of this design is to provide an interchangeable tool interface allowing for the use of various tools for use in a fast paced food preparation area. The target application for his prototype is to demonstrate the creation, preparation, and serving of grill cheese sandwiches.

2 System Overview

The main architecture of the system involves 4 main systems communicating with each other. The control logic system and UR5 system pass information with each other via the network system through an Ethernet TCP/IP connection. The UR5 then provides a signal to the mount system allowing for the docking and un-docking of various utensils.

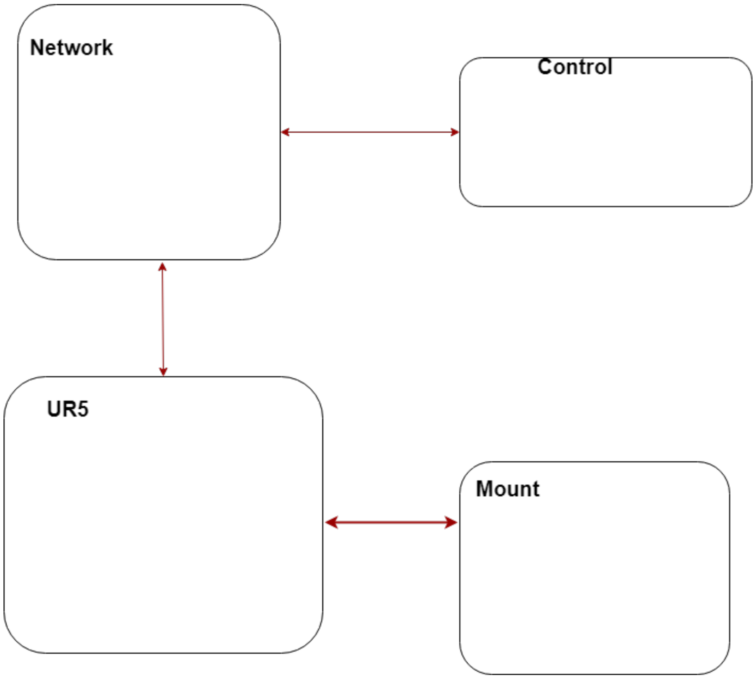


Figure 1: A simple architectural layer diagram

3 Network Layer Subsystems

This layer is for communication between the control layer and the UR5 layer in details. It have a router as its subsystem. The router uses DHCP to allow IP addresses to be assigned to the UR5's control box and the Raspberry Pi.

3.1 Network Layer Hardware

N/A

3.2 Network Layer Operating System

N/A

3.3 Netowrk Layer Software Dependencies

Requires TCP/IP socket connection method in software.

3.4 Router Subsystem

Router is the only piece of hardware in the Network Layer. It requires Ethernet cables to be connected to the UR5 control box and Raspberry Pi establish TCP/IP socket connection for issuing commands to the UR5's robotic arm.

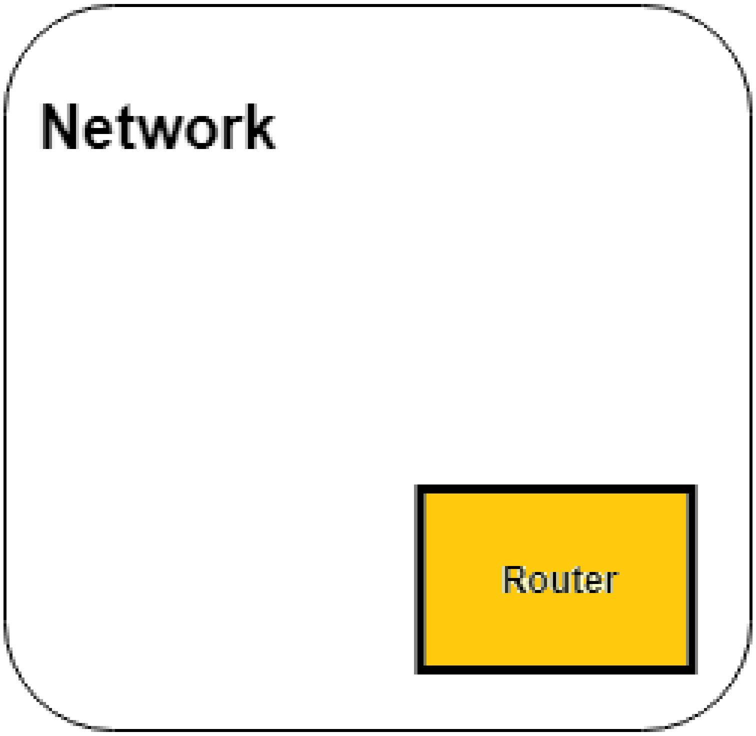


Figure 2: Router Subsystem diagram

3.4.1 Router Subsystem Hardware

Standard wired Ethernet router.

3.4.2 Router Subsystem Operating System

N/A

3.4.3 Router Subsystem Software Dependencies

Requires TCP/IP socket connection method in software.

3.4.4 Router Subsystem Programming Languages

N/A

3.4.5 Router Subsystem Data Structures

Structure of commands issued from Raspberry Pi to UR5 are sent as strings converted to bytes. For example the string 'stop(1.0)' would be converted to bytes and sent to the robot to tell it to stop.

3.4.6 Router Subsystem Data Processing

N/A

4 Control Layer Subsystems

This layer is the integration of the UI and camera via Raspberry Pi as well as the mediator between the user and the UR5. The Node.js environment and OpenCV library are the main software involved in this layer.

4.1 Control Layer Hardware

N/A

4.2 Control Layer Operating System

N/A

4.3 Control Layer Software Dependencies

N/A

4.4 Raspberry Pi

The Raspberry Pi is a piece of hardware that establishes a virtual Node.js server to run the UI for the user interact with. It also communicates with the Network Layer via Ethernet in order to issues commands to the UR5 due to user input.

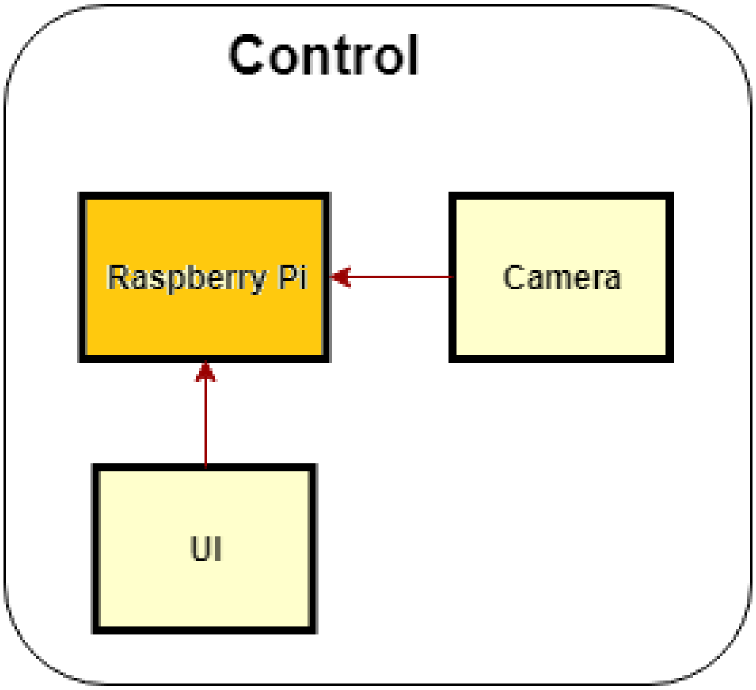


Figure 3: Raspberry Pi subsystem

4.4.1 Raspberry Pi Subsystem Hardware

Raspberry Pi 3 is the main hardware us in this subsystem. It has inputs for connecting the Ethernet cable to the Network Layer and USB and HDMI for connecting to the UI layer (mouse and monitor).

4.4.2 Raspberry Pi Subsystem Operating System

Debian GNU/Linux OS running on Raspberry Pi to initiate code.

4.4.3 Raspberry Pi Subsystem Software Dependencies

OpenCV and Node.js with NPM (Node Package Manager) to ensure all libraries are installed.

4.4.4 Raspberry Pi Subsystem Programming Languages

JavaScript, and HTML are used for establishing the Node.js server. Python 3.6 is used as a means of interpreting and sending URScript commands (Universal Robotics's internal robotic programming language) to the UR5. C++ is used handling camera interactivity with the OpenCV library (Not fully implemented).

4.4.5 Raspberry Pi Subsystem Data Structures

Structure of commands issued from Raspberry Pi to UR5 are sent as strings converted to bytes. For example the string 'stop(1.0)' would be converted to bytes and sent to the robot to tell it to stop.

4.4.6 Raspberry Pi Subsystem Data Processing

N/A

4.5 Camera (Not fully implemented)

The camera subsystem provide a video feed to the Raspberry Pi for image processing.

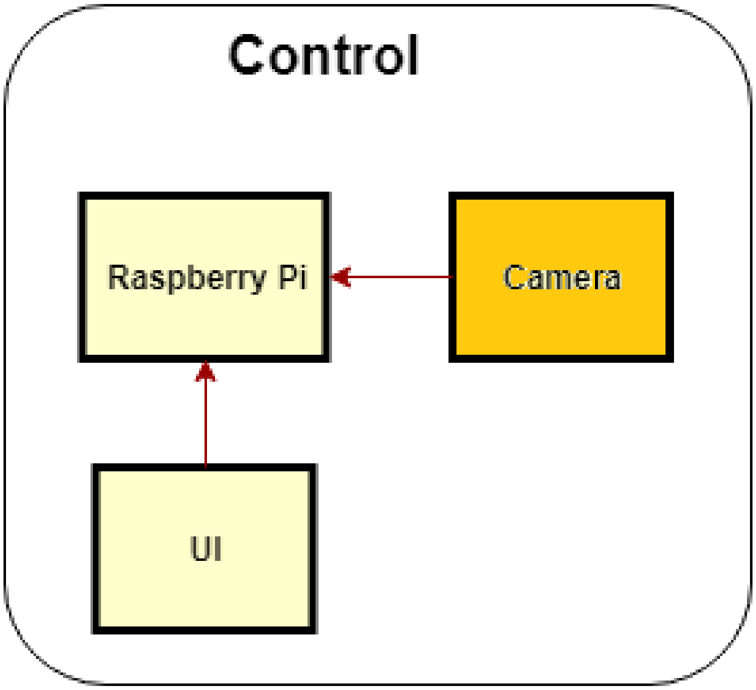


Figure 4: Camera subsystem

4.5.1 Raspberry Pi Subsystem Hardware

The camera itself is a 720p web cam.

4.5.2 Raspberry Pi Subsystem Operating System

N/A

4.5.3 Raspberry Pi Subsystem Software Dependencies

N/A

4.5.4 Raspberry Pi Subsystem Programming Languages

N/A

4.5.5 Raspberry Pi Subsystem Data Structures

N/A

4.5.6 Raspberry Pi Subsystem Data Processing

N/A

4.6 User Interface (UI)

The UI is the main interface the user uses to issues commands to the UR5 robotic arm.

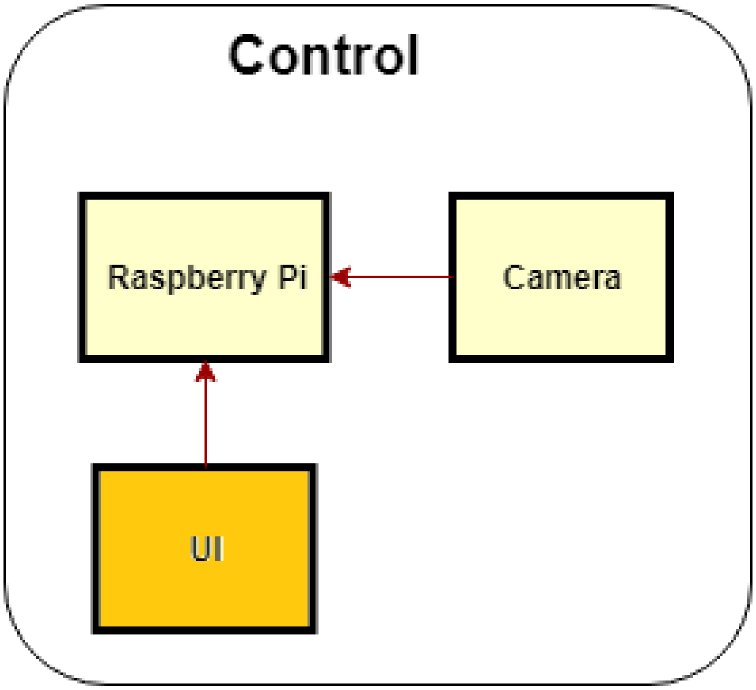


Figure 5: UI subsystem

4.6.1 UI Subsystem Hardware

This subsystem uses a monitor and mouse connected to the Raspberry Pi and displays a UI with several buttons for the user to click in order to tell the UR5 what do.

4.6.2 UI Subsystem Operating System

N/A

4.6.3 UI Subsystem Software Dependencies

A virtual Node.js server with JavaScript as back-end of the UI is required.

4.6.4 UI Subsystem Programming Languages

JavaScript, and HTML are used for establishing the Node.js server.

4.6.5 UI Subsystem Data Structures

N/A

4.6.6 UI Subsystem Data Processing

N/A

5 UR5 Layer Subsystems

This layer is the destination of all the instructions from the Control Layer go to. It contains the UR5's arm, its control box, and a Polyscope for the UI.

5.1 UR5 Layer Hardware

The UR5 is a sophisticated robot system developed and created by Universal Robotics. It contains a 6-jointed robotic arm that sends and receives data to control box. The control box itself sends and receives commands through the Polyscope, a large hand-held touch pad device.

5.2 UR5 Layer Operating System

N/A

5.3 UR5 Layer Software Dependencies

N/A

5.4 UR5 Arm

This subsystem consists of a 6-jointed arm that houses a servo for each of those joints that can rotate 360 degrees in either direction. It receives signals from the control box to determine what pose should be in.

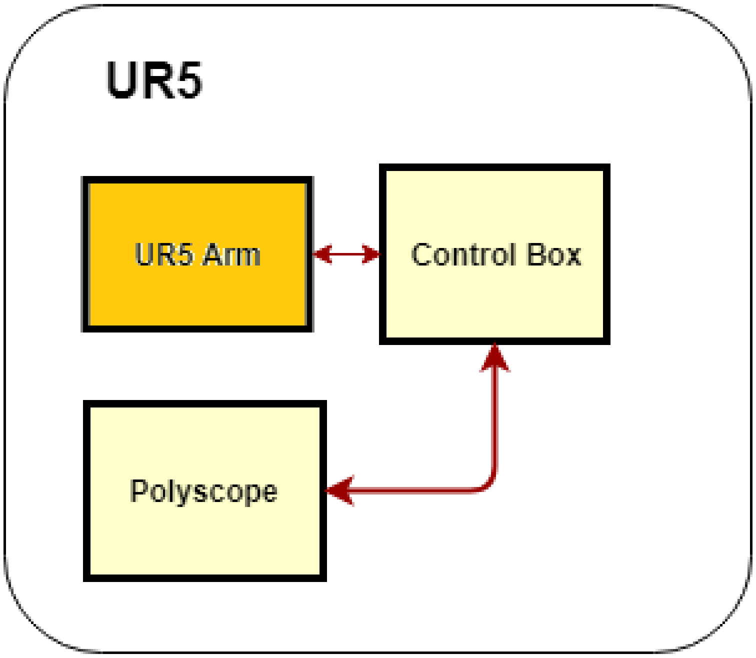


Figure 6: UR5 Arm subsystem diagram

5.4.1 UR5 Arm Subsystem Hardware

It contains servos and a metal housing for each "limb" of the arm that holds most of the wiring. The first limb is mounted to a table and the final limb has most components in the Mount Layer attached to it.

5.4.2 UR5 Arm Subsystem Operating System

N/A

5.4.3 UR5 Arm Subsystem Software Dependencies

N/A

5.4.4 UR5 Arm Subsystem Programming Languages

N/A

5.4.5 UR5 Arm Subsystem Data Structures

N/A

5.4.6 UR5 Arm Subsystem Data Processing

N/A

5.5 Control Box

A central processing area for the UR5 Layer. This houses a computer that runs a Linux OS with a proprietary UI powered by Java. This UI is used and managed in the Polyscope.

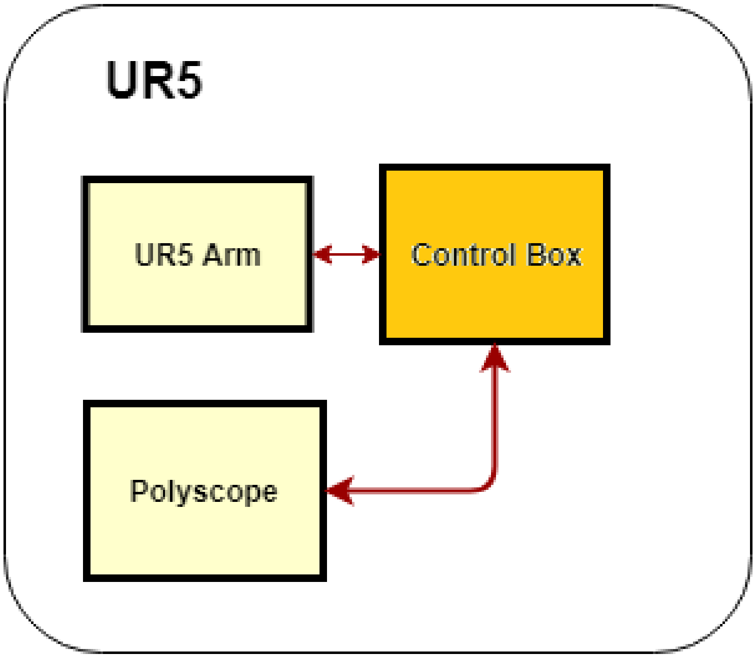


Figure 7: UR5 Control Box subsystem diagram

5.5.1 UR5 Arm Subsystem Hardware

It contains servos various computing equipment (motherboard, hard drive, etc.)

5.5.2 Subsystem Operating System

This subsystem runs Linux.

5.5.3 Subsystem Software Dependencies

A Java Runtime Environment of at least 6.0 is required to run the UI displayed on the Polyscope.

5.5.4 Subsystem Programming Languages

N/A

5.5.5 Subsystem Data Structures

N/A

5.5.6 Subsystem Data Processing

N/A

5.6 Polyscope

A large touch pad that serves as the main UI for a user when developing for the UR5.

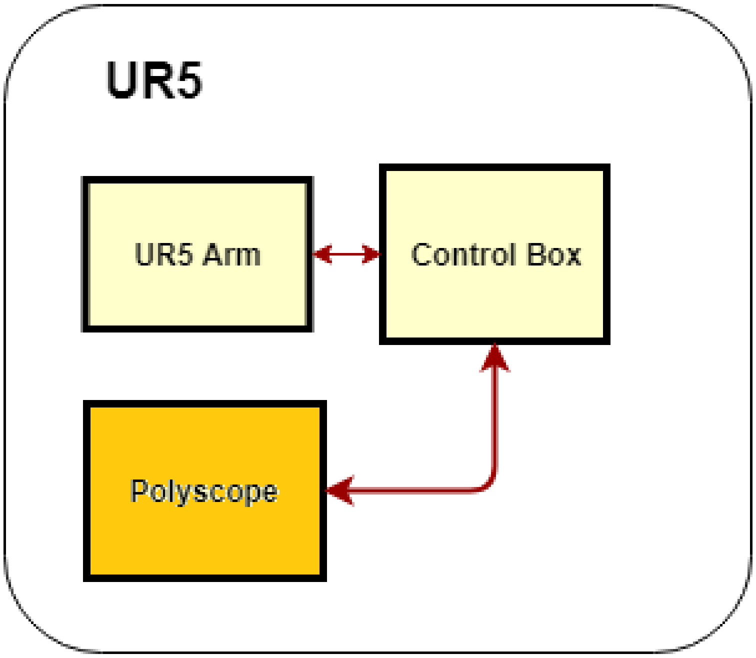


Figure 8: UR5 Control Box subsystem diagram

5.6.1 UR5 Arm Subsystem Hardware

It contains a touch sensitive screen as well as a cable made solely for connecting to the UR5's control box. A emergency stop button is also located on the pad for safety reasons.

5.6.2 Subsystem Operating System

N/A

5.6.3 Subsystem Software Dependencies

N/A

5.6.4 Subsystem Programming Languages

N/A

5.6.5 Subsystem Data Structures

N/A

5.6.6 Subsystem Data Processing

N/A

6 Mount Layer Subsystems

This layer is the mount that controls the release and locking of the tool.

6.1 Mount Layer Hardware

There is a 3D printed housing for this layer using PLA plastic. It contains the magnet as well as a insertion point for the tool being used by the robotic arm. A wire connecting the magnet to the UR5's arm is located here. (There is an internal wire that connects to the control box). The tool is assumed to have a metal contact for the magnet to attach to.

6.2 Mount Layer Operating System

N/A

6.3 Mount Layer Software Dependencies

N/A

6.4 Connector

This interface is an 8 pin connection point that uses a Lumberg RKMV 8-354 cable. This is how the control box sends a signal to the magnet to provide it power.

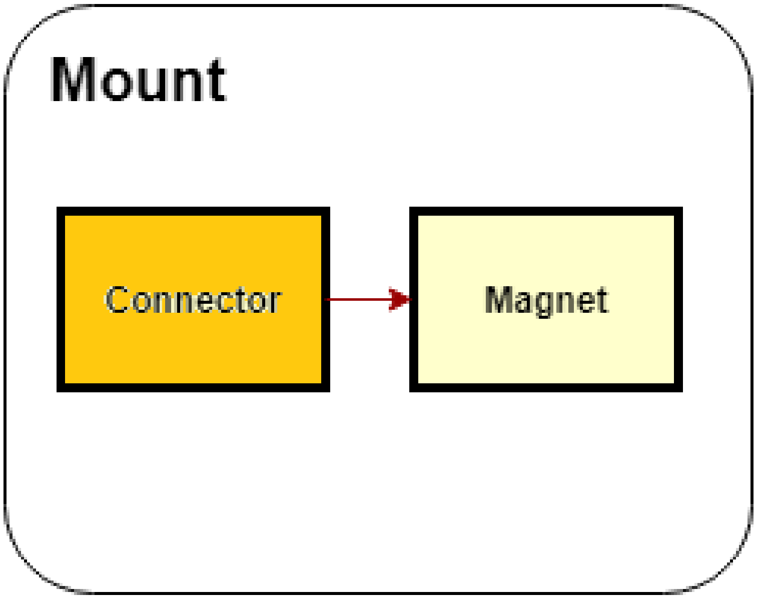


Figure 9: Mount Layer Connector diagram

6.4.1 Connector Subsystem Hardware

The 8 pin connection is standard connection in robots uses to provide various signals to connected equipment. In this case a 12V voltage drop is provided to the magnet.

6.4.2 Connector Subsystem Operating System

N/A

6.4.3 Connector Subsystem Software Dependencies

N/A

6.4.4 Subsystem Programming Languages

N/A

6.4.5 Connector Subsystem Data Structures

N/A

6.4.6 Subsystem Data Processing

N/A

6.5 Magnet

This is the subsystem that makes contact directly with the tool being used and locks or releases it depending on signals from the connector subsystem.

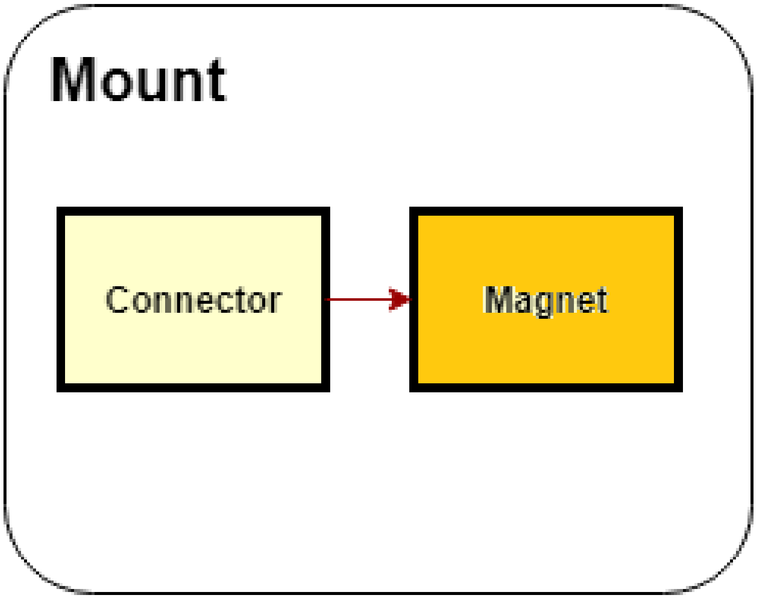


Figure 10: Mount Layer Connector diagram

6.5.1 Connector Subsystem Hardware

The 12V magnet of this system has a payload of about 11 pounds and is the contact point for the metallic end of the tool's handle/shaft.

6.5.2 Connector Subsystem Operating System

N/A

6.5.3 Connector Subsystem Software Dependencies

N/A

6.5.4 Subsystem Programming Languages

N/A

6.5.5 Connector Subsystem Data Structures

N/A

6.5.6 Subsystem Data Processing

N/A

7 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.