Reliability and Safety Analysis

Year: 2023 Semester: Spring Team: 3 Project: "Rigged" Card Shuffler
Creation Date: 3/31/23 Last Modified: 4/1/23
Author: Utkarsh Priyam Email: upriyam@purdue.edu

Assignment Evaluation:

Item	Score (0-5)	Weight	Points	Notes
Assignment-Specific Items				
Reliability Analysis		x2		
MTTF Tables		x3		
FMECA Analysis		x2		
Schematic of Functional Blocks (Appendix A)		x2		
FMECA Worksheet (Appendix B)		x3		
Writing-Specific Items				
Spelling and Grammar		x2		
Formatting and Citations		x1		
Figures and Graphs		x2		
Technical Writing Style		x3		
Total Score				

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

Comments:
Comments from the grader will be inserted here.

1.0 Reliability Analysis

The components in our design that are most likely to fail are our STM32 microcontroller and our Raspberry Pi's BCM2837 microprocessor, which are high complexity ICs, alongside our three components that run at high temperatures due to power handling: our 12V to 5V step down voltage regulator (D36V28F5), the 5V to 3.3V voltage regulator (LD117), and our DC motor power control BJTs (TIP41). The former two were chosen as they are the most complex components in our product, and as a result they are the most likely to fail probabilistically. The latter three were chosen as they handle high voltage power transformation and control, which results in a lot of heat flowing through those components. This makes them more likely to fail compared to other components in our design.

For the failure rate computations, we used the formula $\lambda_P = (C_1 \pi_T + C_2 \pi_E) \pi_Q \pi_L$ [1] taken from the provided Military Handbook on page 5-3, which gives the failure rate as failures per 10^6 hours.

The formula to convert λ_P to mean time to failure (MTTF) is MTTF = $10^6 / (24 * 365 * \lambda_P)$ years.

For the failure rate formula, the π_E, π_Q, π_L coefficients are fixed for all components. This is true for π_E as all components will be operating in the same environment. This is true for π_Q as all components selected are commercial products rather than those necessarily certified and validated for military use, so they use a common quality rating for commercial products. Finally, this is true for π_L because all of the components used are ones that have existed for years, if not decades. The learning coefficient caps off after 2 years, so they all share the same π_L value. Specifically, we will be using $\pi_E = 4.0$ for G_M (ground, mobile) [1], which indicates that the product can be used in any ground-based environment without any dedicated environmental control for factors such as temperature and humidity. We will use $\pi_Q = 10$ for unspecified commercial components [1]. Lastly, we will use $\pi_L = 1.0$ for devices that have been in production for at least 2 years [1].

For π_T, we consider the STM32 and BCM2837 microcontrollers to have expected peak operating temperatures of no more than 85 °C, which would result in $\pi_T = 0.98$ [1]. The two power regulators have expected peak operating temperatures of 125 °C, which results in a value of π_T = 3.1 [1]. Finally, the motor control BJT has a peak operating temperature of 150 °C, for a final value of $\pi_T = 180$ [1].

The full component breakdown and calculations are tabulated below:

STM32 Microcontroller [2]

Parameter name	**Description**	**Value**	**Comments** regarding choice of parameter value, especially if you had to make assumptions.
C_1	Die complexity	0.56	32-bit processor [1]
π_T	Temperature coeff.	0.98	Identified above
C_2	Package failure rate	0.025	64 pin SMT [1]
π_E	Environment coeff.	4.0	Identified above

π_Q	Quality coeff.	10	Identified above
π_L	Learning coeff.	1.0	Identified above
λ_P	Failure rate (f/10^6 hrs)	6.49	
MTTF	MTTF (yrs * units)	17.59	**17.59 yrs** overall

BCM2837 Microprocessor (from Raspberry Pi) [3]

Parameter name	**Description**	**Value**	**Comments** regarding choice of parameter value, especially if you had to make assumptions.
C_1	Die complexity	1.12	64-bit processor [1]
π_T	Temperature coeff.	0.98	Identified above
C_2	Package failure rate	0.053	128 pin SMT [1]
π_E	Environment coeff.	4.0	Identified above
π_Q	Quality coeff.	10	Identified above
π_L	Learning coeff.	1.0	Identified above
λ_P	Failure rate (f/10^6 hrs)	13.1	
MTTF	MTTF (yrs * units)	8.71	**8.71 yrs** overall

D36V28F5 voltage regulator [4]

Parameter name	**Description**	**Value**	**Comments** regarding choice of parameter value, especially if you had to make assumptions.
C_1	Die complexity	0.020	[1]
π_T	Temperature coeff.	3.1	Identified above
C_2	Package failure rate	0.012	32 pin SMT [1]
π_E	Environment coeff.	4.0	Identified above
π_Q	Quality coeff.	10	Identified above
π_L	Learning coeff.	1.0	Identified above
λ_P	Failure rate (f/10^6 hrs)	1.1	
MTTF	MTTF (yrs * units)	103.78	**103.78 yrs overall**

LD117 voltage regulator [5]

Parameter name	**Description**	**Value**	**Comments** regarding choice of parameter value, especially if you had to make assumptions.
C_1	Die complexity	0.010	[1]
π_T	Temperature coeff.	3.1	Identified above
C_2	Package failure rate	0.0013	4 pin SMT [1]
π_E	Environment coeff.	4.0	Identified above
π_Q	Quality coeff.	10	Identified above
π_L	Learning coeff.	1.0	Identified above
λ_P	Failure rate (f/10^6 hrs)	0.362	

MTTF	MTTF (yrs * units)	315.3	**315.3 yrs overall**

TIP41 BJT [6] (x2 units)

Parameter name	**Description**	**Value**	**Comments** regarding choice of parameter value, especially if you had to make assumptions.
C_1	Die complexity	0.010	1, Linear [1]
π_T	Temperature coeff.	180	Identified above
C_2	Package failure rate	0.00092	3 pin through-hole [1]
π_E	Environment coeff.	4.0	Identified above
π_Q	Quality coeff.	10	Identified above
π_L	Learning coeff.	1.0	Identified above
λ_P	Failure rate (f/10^6 hrs)	18.0	
MTTF	MTTF (yrs * units)	6.34	**3.17 yrs overall**

Overall, the voltage regulators are the most reliable, with MTTFs of over 100 years each. The two microprocessors fall at around 1-2 decades each, which is still more than reasonable for a niche consumer product. However, the BJTs are problematic, as with two units in the device the MTTF for the pair comes down to just over 3 years. In the long run, this may end up being one of the most-serviced components in the device, due to the high operating temperatures.

Regarding refinements that could improve the design's reliability, the primary locations of focus would be the BJTs and microprocessors. The former could be improved if lower operating temperatures could be assured, either by limiting voltage/current drawn by the motors or by applying cooling solutions to the BJTs. The microprocessors could be made more reliable by under-spec-ing them to the bare minimum complexities and package sizes. For instance, the SBC could very well suffice with a 32-bit computer instead of a 64-bit Raspberry Pi. However, the easiest solution to improving reliability across the board would be to use components that have been tested and certified for increased quality (i.e., lower π_Q), such as by using military-grade components rather than commercial ones.

2.0 Failure Mode, Effects, and Criticality Analysis (FMECA)

Our product schematic can be divided into the following 5 functional blocks: power circuitry, microcontroller, Raspberry Pi interface, user interface, and motor control. These blocks are pictured in the same order in Appendix A.

For the power circuitry, the potential failure conditions are voltage surges, brownouts, or possibly even complete blackouts. The former can be caused by a failure of either voltage regulator, which results in voltages outside the specified range (either 5V or 3.3V) being supplied. The latter two can result from a variety of issues, including regulator failures, short circuits near the power circuitry or throughout the rest of the board, or even failure of the bypass capacitor. In the case of a voltage surge, many components on and off the board, including resistors, capacitors, the STM32 microcontroller, and the Raspberry Pi, can potentially be damaged or destroyed. Brownouts could result in inconsistent signaling or motor control, which could result in inaccurate logical function of the product. Blackouts could result in loss of state, product shutdown, and similar nondestructive failures. In all three cases, failures would only be observable by a user once the product either ceases to function or downstream effects manifest in user-interactable subsystems such as the user interface.

The microcontroller can experience various potential failure conditions, mostly related to power issues mentioned above. Outside of those, potential failure states include complete reset or failure to continue instruction execution. The former could be caused by a short circuit within the reset circuitry and the latter by various issues for the microcontroller, including incorrect code, poor operating conditions including temperature and humidity, or even power fluctuations that result in invalid states. These errors cannot be observed by a user unless it results in downstream effects, such as the entire system ceasing operation due to lack of response or control by the microcontroller on the various motors, buttons, and display.

The Raspberry Pi interface can fail by failing to deliver power to the RasPi, incorrectly streaming data over UART (either misformatting the UART bits or the packet bytes), or dropping power to the camera LED required for card illumination. All three would result from misconfiguration of the microcontroller, either in software or as a result of the aforementioned microcontroller failure states. These failures would be completely invisible to the user, as the product could theoretically continue to function, albeit with incorrect logic, with decreased illumination or scrambled data exchange between the RasPi and the microcontroller.

The user interface could fail by incorrectly passing inputs to the microcontroller and incorrectly displaying outputs on the LCD panel. These failures could be related to component-specific issues, such as button and resistor failures, or GPIO pin failures on the microcontroller. These issues would be immediately obvious to the user as inputs would be dropped or the display would either freeze or become blank following such a failure state.

Lastly, the motor control systems could fail by failing to turn the motors at all or locking them in a permanently on state. The former could be the result of component burnout, such as the BJTs, the microcontroller's GPIO pins, or even resistors and capacitors. The latter could result from BJT failure or short circuits. The former would be somewhat transparent to the user, until he or she realizes that the cards are not being ingested and/or outputted. The latter could be identified by the incessant motor whirring noise, but would have little to no effect on operation, as the DC motors would typically remain on for the duration of shuffling anyways.

Regarding failure criticality, we define three levels of failure: low, medium, and high. Low criticality failures refer to those which may affect the product's operation, but where failures are limited to the failed components and possibly a few others within the functional block. Medium criticality refers to those which still have no impact on the user, other than potential user experience issues, but which could result in elevated levels of damage throughout the product, including across functional blocks. Finally, high criticality failures refer to those which result in extensive damage within the product, as well as any failures that result in harm to the user. For these failure modes, low criticality failures have an acceptable failure rate of roughly $\lambda < 10^{-6}$, medium failures are acceptable below $\lambda < 10^{-7}$, and high failures are acceptable only for $\lambda < 10^{-9}$.

3.0 Sources Cited:

[1] "Military Handbook: Reliability Prediction of Electronic Equipment," 1990. [Online]. Available: https://purdue.brightspace.com/d2l/le/dropbox/703798/730655/DownloadAttachment?fid=25381497. [Accessed: 1-Apr-2023].

[2] "STM32F091xB, STM32F091xC Datasheet," 2017. [Online]. Available: https://engineering.purdue.edu/477grp3/Files/refs/uc_datasheet.pdf. [Accessed: 1-Apr-2023].

[3] "Raspberry Pi 3 Model B Datasheet," 2018. [Online]. Available: https://engineering.purdue.edu/477grp3/Files/refs/sbc_datasheet.pdf. [Accessed: 1-Apr-2023].

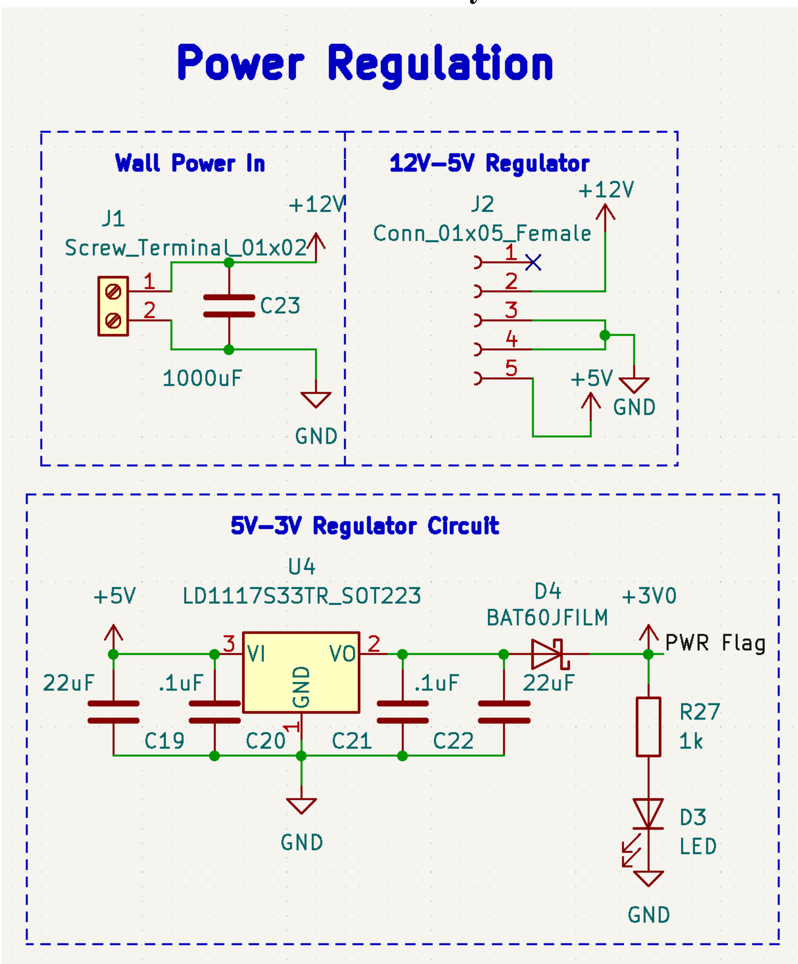
[4] "5V, 3.2A Step-Down Voltage Regulator D36V28F5." [Online]. Available: https://engineering.purdue.edu/477grp3/Files/refs/5v_step_down_regulator.pdf. [Accessed: 1-Apr-2023].

[5] "LD1117 Datasheet," 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/ld1117.pdf. [Accessed: 1-Apr-2023].

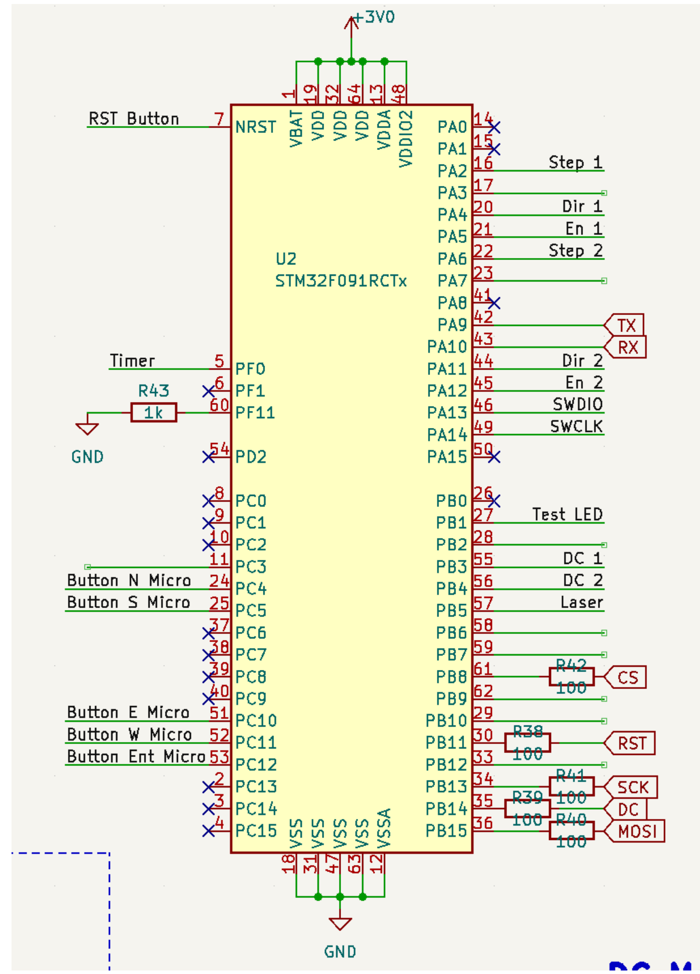
[6] "TIP41A / TIP41B / TIP41C NPN Epitaxial Silicon Transistor Datasheet," 2017. [Online]. Available: https://engineering.purdue.edu/477grp3/Files/refs/motor_bjt.pdf. [Accessed: 1-Apr-2023].

Appendix A: Schematic Functional Blocks

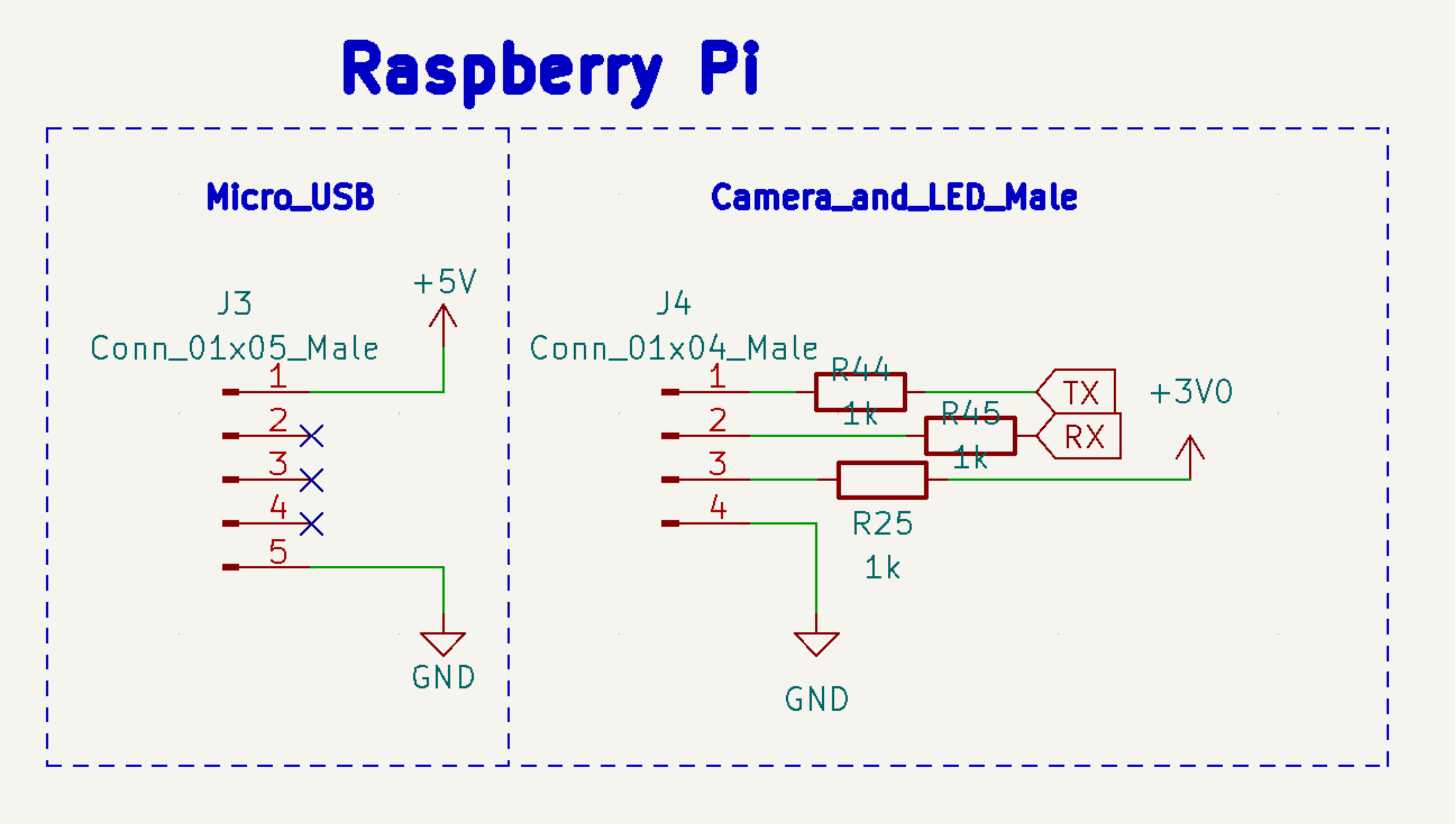
Power Circuitry



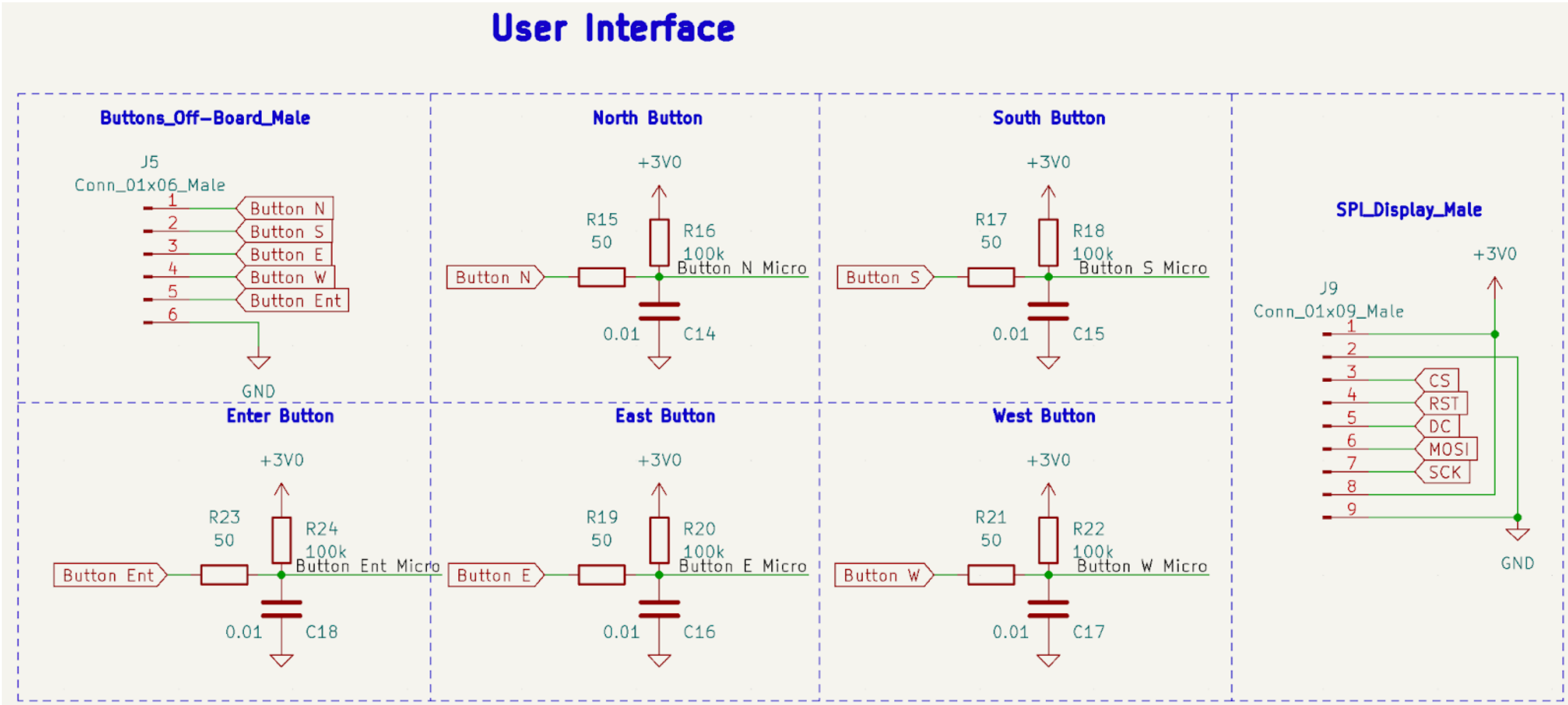
Microcontroller



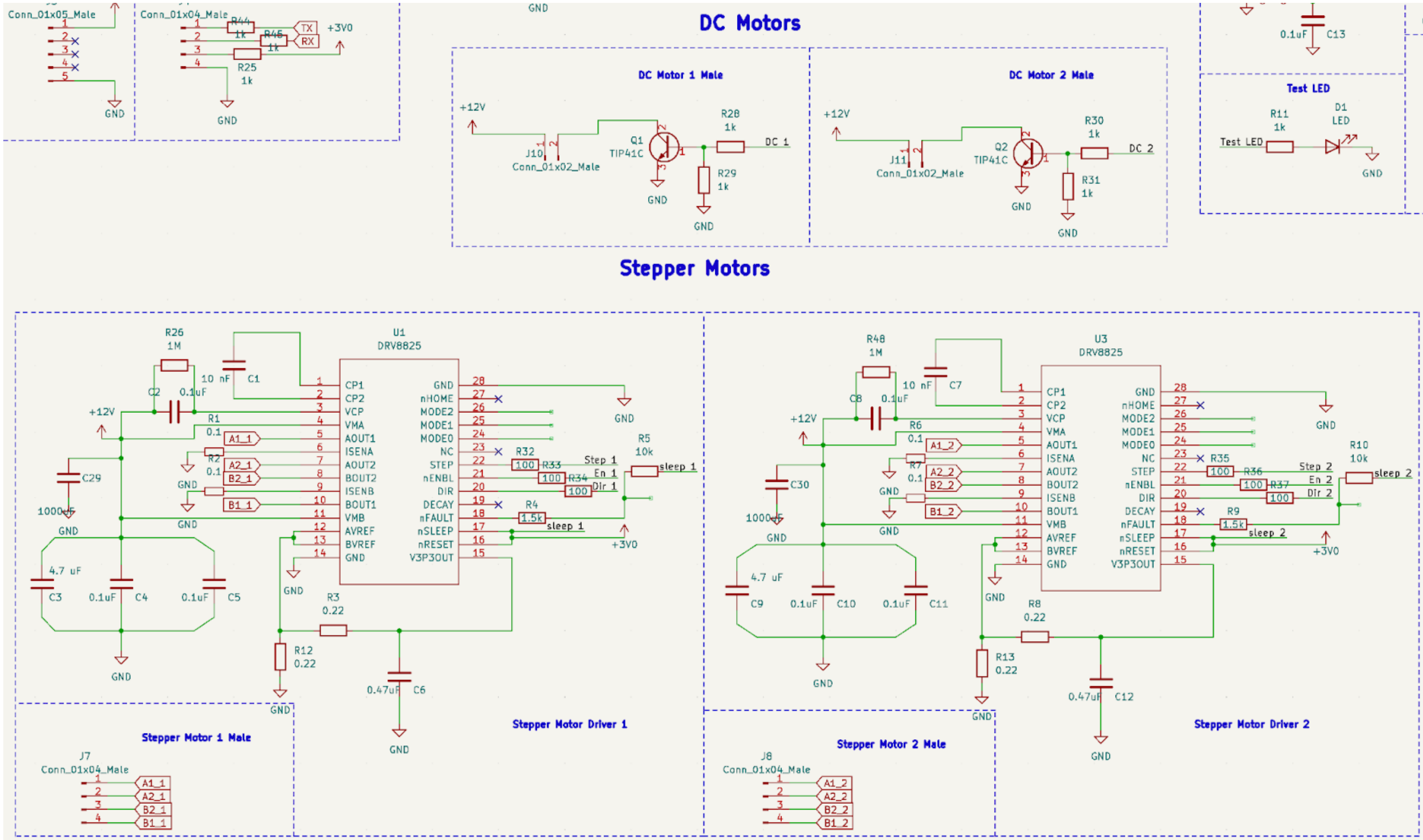
Raspberry Pi Interface



User Interface



Motor Control



Appendix B: FMECA Worksheet

Subsystem A: Power Circuitry

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
A1	3.3V output is too high	Either voltage regulator failed, or line shorted with higher power line	Microcontroller is damaged	None without opening product and/or downstream effects	Medium	
A2	3.3V output is too low	Either voltage regulator failed, or line was short circuited, or bypass capacitor failed	Components fail to operate at specified frequencies or at all	None without opening product and/or downstream effects	Low	
A3	5V output is too high	Voltage regulator failed, or line shorted with higher power line	Raspberry Pi and/or microcontroller is damaged	None without opening product and/or downstream effects	Medium	
A4	5V output is too low	Voltage regulator failed, or line was short circuited, or bypass capacitor failed	Components fail to operate at specified frequencies or at all	None without opening product and/or downstream effects	Low	
A5	12V output is too high	12V power supply failed and/or external AC power source surged	Motors, motor drivers, Raspberry Pi, and/or microcontroller are damaged	None without opening product and/or downstream effects	Medium	

A6	12V output is too low	12V power supply failed, or barrel jack connection is lose	Components fail to operate at specified frequencies or at all	None without opening product and/or downstream effects	Low	

Subsystem B: Microcontroller

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
B1	Microcontroller reset	Reset circuitry short circuited	Product stops operation	None except through downstream effects	Low	
B2	Failure to execute instructions	Microcontroller breaks, or is run under poor operating conditions (i.e. temperature or humidity)	Product stops operation	None except through downstream effects	Low	

Subsystem C: Raspberry Pi Interface

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
C1	Low/No power to Raspberry Pi	Voltage supply issues or loose cable	RasPi does not activate/fails to operate at speed	None except through downstream effects	Low	

C2	Low/No power to camera LED	Voltage supply issues or loose cable or GPIO pin issues	Poor lighting for camera → incorrect card recognition	None, except opening product during operation	Low	Has absolutely no impact on product operation, except vis-a-vis outputting cards in the specified “rigged” order
C3	Incorrect communication with microcontroller	GPIO pin failure, incorrect code, or loose cables	Loss of program state, deadlock, or incorrect shuffle output	None, except for analyzing electrical data or monitoring shuffle output	Low	

Subsystem D: User Interface

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
D1	Incorrect inputs from button input array	Broken components (resistors, buttons, capacitors) or broken GPIO pins	Inputs not registered correctly	LCD screen does not update as expected for attempted input	Low	
D2	Incorrect output displayed by LCD panel	Broken components (LCD, resistors) or broken GPIO pins	Outputs not displayed correctly	Garbage display output, or completely blank screen	Low	

Subsystem E: Motor Control

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks

E1	DC motor fails to spin	BJT breaks, or GPIO pin fails, or power is short circuited	Card ingest stops working	Cards are not removed from input, product stops operation	Low	
E2	DC motor spins permanently	BJT breaks, or GPIO pin fails, or power is short circuited	NONE	Incessant whirring when product is powered but not in operation	Medium	Excess heat and/or noise pollution could cause issues
E3	Stepper motor fails to spin	Motor driver breaks, or GPIO pin fails, or power is short circuited	Card ingest or sorting stops working	Card input and/or output fails, or cards get stuck in product	Low	
E4	Stepper motor spins with wrong bin size	Motor driver breaks, or hardwired resistor connections fail	Card sorting/binning fails to sort properly, but cards are still outputted in some meaningless, semi-random order	Cards are not outputted in expected or specified order	Low	Depending on step size, cards may get slightly stuck, or stack up, affecting the sorting process but without completely breaking the product