RPI Spida V0.3.3 Control System Guide

This document describes the combined command logic for the RPI Spida robot, which utilizes **Card Detection (Primary Control)** and **Filtered Motion Detection (Safety Override)** to determine movement.

1. Core Logic & Priority System

The Spida controller operates on a clear, two-tiered priority system:

Priority Level	Source	Purpose	Action
P0 (Highest)	**9 of Diamonds (9D)**	**System Crash**	Immediate and unrecoverable program termination (sys.exit(1)).
P1 (High)	**Card Command**	Primary Movement	Executes movement, rotation, or halt based on the detected playing card.
P2 (Override)	**Large Motion**	Safety Stop	Overrides any P1 command (except 9D) and enforces an immediate **STOP**.
P3 (Lowest)	**Small Motion**	Visualization	Draws visual cues (blue circles) and counts small objects, but does **not** affect movement.

2. Card Command Mapping (P1)

Card detection uses specific ranks and suits to issue movement commands. Commands for

Aces (Rotation) take precedence over general suit commands (Strafe/Move).

Card / Suit	**Command**	**Movement Type**	**Description**	**Example Card**
9D	**CRITICAL CRASH**	N/A	Immediately terminates the entire control program.	9 of Diamonds
Ace of Clubs	**TURN LEFT**	Rotation	Rotates the robot counter-clockwise in place.	**AC**
Ace of Spades	**TURN RIGHT**	Rotation	Rotates the robot clockwise in place.	**AS**
Diamonds (D)	**FORWARD**	Drive	Moves the robot straight in its current facing direction.	**KD**, **7D**
Hearts (H)	**BACKWARD**	Drive	Moves the robot straight backwards.	**KH**, **7H**
Clubs (C)	**MOVE LEFT**	Strafe	Strafe movement **left**, relative to the robot's current facing direction.	**KC**, **7C** (not AC)
Spades (S)	**MOVE RIGHT**	Strafe	Strafe movement **right**, relative to the robot's current facing direction.	**KS**, **7S** (not AS)

3. Motion Detection Logic (P2 & P3)

The system constantly analyzes the video stream for motion artifacts and categorizes them by size (area) to decide the appropriate response.

Motion Category	**Area Range (Pixels)**	**Response**	**Effect on Spida Movement**
Noise	$< 50$	Ignored	None.
Small Item (P3)	$50 \leq Area < 5,000$	**Visualization/Count**	Draws a **blue circle** and increments the counter. **Movement continues.**
Safety Stop (P2)	$Area \geq 99,999$	**Safety Override**	Triggers an immediate **STOP_MOTION** command.

Key Feature: The SMALL ITEM category allows the robot to continue executing its primary card command while still providing visual feedback and a count of minor disturbances in the environment.

4. Step-by-Step Command Examples

Here's how the system determines the final action based on various scenarios:

Scenario 1: Basic Movement

1. **Input:** A **King of Hearts (KH)** is placed in the camera view.
2. **Card Logic (P1):** Detects 'H' (Hearts) and issues the command **BACKWARD**.
3. **Motion Logic (P2):** No significant motion is detected.
4. **Final Action:** The Spida executes **BACKWARD** movement.

Scenario 2: Rotation

1. **Input:** An **Ace of Clubs (AC)** is placed in the camera view.
2. **Card Logic (P1):** Detects 'AC' and issues the command **TURN LEFT**.
3. **Motion Logic (P2):** No significant motion is detected.
4. **Final Action:** The Spida executes **TURN LEFT** rotation.

Scenario 3: Safety Override

1. **Input:** A **5 of Diamonds (5D)** is placed in the camera view, **AND** a large obstacle (Area $\geq 99,999$) suddenly enters the frame.
2. **Card Logic (P1):** Detects 'D' (Diamonds) and issues the command **FORWARD**.
3. **Motion Logic (P2):** Detects large motion. The safety override flag is set.
4. **Final Action:** The P2 override takes control, and the Spida executes **STOP_MOTION**. (Movement stops and the simulator turns red).

Scenario 4: Multiple Cards (Rotation Priority)

1. **Input:** An **Ace of Spades (AS)** and a **King of Diamonds (KD)** are placed in the camera view simultaneously.
2. **Card Logic (P1):**
    - Detects 'AS' (TURN RIGHT)
    - Detects 'D' (FORWARD)
    - **Priority Rule:** Rotation commands (Aces) override general suit commands.
3. **Final Action:** The Spida executes **TURN RIGHT**.

Scenario 5: Critical Crash

1. **Input:** The **9 of Diamonds (9D)** is placed in the camera view.
2. **Card Logic (P0):** Detects '9D'.
3. **Final Action:** The program prints a critical error message, flushes all logs, and immediately **crashes/exits** the process.