**Arduino Dome Automation Controller**

This Arduino code controls a motorized astronomical dome using relays, limit switches, an ultrasonic sensor, and magnetic limit switches. The dome has two overlapping sections (West and East) that open and close in a specific sequence to ensure weatherproofing.

**Hardware Components**

* **Arduino Board:** The microcontroller running the code.
* **Relay Modules:** Two dual-relay modules to control the West and East side motors.
* **Limit Switches:**
  + West Open Limit Switch
  + West Closed Limit Switch
  + East Open Limit Switch
  + East Closed Limit Switch
* **Ultrasonic Sensor (HC-SR04):** To detect if the telescope mount is parked.
* **Magnetic Limit Switches:**
  + West Partial Open Switch
  + East Partial Close Switch
* **DC Motors:** Two DC motors to drive the West and East dome sections.

**Wiring Diagram**

(Include the image you provided here: "Reversible DC Motor Using 2 SPDT Relays")

**Pin Definitions**

// Define pins

const int relayPin\_west\_1 = 2;

const int relayPin\_west\_2 = 3;

const int relayPin\_east\_1 = 4;

const int relayPin\_east\_2 = 5;

const int limitSwitch\_west\_openPin = 6;

const int limitSwitch\_west\_closedPin = 7;

const int limitSwitch\_east\_openPin = 8;

const int limitSwitch\_east\_closedPin = 9;

// Define ultrasonic sensor pins

const int trigPin = 10; // Ultrasonic sensor trigger pin

const int echoPin = 11; // Ultrasonic sensor echo pin

// Define magnetic limit switch pins

const int westPartialOpenPin = 12; // Pin for west side partial open switch

const int eastPartialOpenPin = 13; // Pin for east side partial open switch (if needed)

Actuator States

enum ActuatorState { OPEN, CLOSED, OPENING, CLOSING, STOPPED };

// Actuator State Variables

ActuatorState actuator\_west\_State = STOPPED;

ActuatorState actuator\_east\_State = STOPPED;

**Code Explanation**

1. **Initialization (setup()):**
   * Sets the pin modes for relays, limit switches, ultrasonic sensor, and magnetic switches.
   * Initializes serial communication for debugging and status reporting.
2. **Main Loop (loop()):**
   * **Command Processing:**
     + Listens for commands from the serial port (e.g., "OPENSHUTTER," "CLOSESHUTTER").
     + Validates commands and sets the appropriate flags (openingInProgress, closingInProgress).
     + Checks if the telescope is parked (using the ultrasonic sensor) before allowing the dome to close.
   * **Actuator Control Logic:**
     + **Opening Sequence:**
       - The West side starts opening first.
       - Once the West side reaches a partial open position (detected by the westPartialOpenPin), the East side starts opening.
       - Both sides continue opening until their respective limit switches are triggered.
     + **Closing Sequence:**
       - The East side closes first.
       - Once the east side reaches a partial close position (detected by the eastPartialOpenPin), the east side continues closing until fully closed.
       - Once the East side is fully closed, the West side starts closing.
       - Both sides continue closing until their respective closed limit switches are triggered.
   * **Sensor Reading:**
     + Uses an ultrasonic sensor to determine if the telescope mount is parked.
     + Uses limit switches to detect the open and closed positions of the West and East sides.
     + Uses magnetic limit switches to detect the partial open and close positions of the west and east sides.
   * **Status Reporting:**
     + Sends status messages to the serial port at regular intervals, indicating the dome's state (opening, closing, open, closed, stopped, parked).
   * **Debouncing:**
     + Debounces the limit switch inputs to prevent false triggering.
   * **Ultrasonic sensor:**
     + Measures distance and compares it to a threshold to see if the scope is parked.
3. **Functions:**
   * openActuator(), closeActuator(), stopActuator(): Control the relay outputs to move the dome sections.
   * readDebouncedPin(): Debounces digital input pins.
   * measureDistance(): Measures distance using the ultrasonic sensor.
   * isTelescopeParked(), isWestOpen(), isWestClosed(), isEastOpen(), isEastClosed(): Read sensor inputs.
   * sendStatusToDriver(): Sends status messages to the serial port.

**Key Features**

* **Overlapping Dome Control:** Ensures proper opening and closing sequence for overlapping dome sections.
* **Parked Detection:** Prevents closing the dome when the telescope is not parked.
* **Limit Switch Protection:** Uses limit switches to prevent over-travel.
* **Magnetic Limit Switch Synchronization:** Uses magnetic limit switches to ensure the proper timing of the opening and closing sequence.
* **Ultrasonic sensor:** uses an ultrasonic sensor to detect the parking position of the telescope.
* **Serial Communication:** Allows for remote control and status monitoring.

**Usage**

1. Upload the code to your Arduino board.
2. Connect the hardware components as described in the wiring diagram.
3. Open the Arduino Serial Monitor to send commands and view status messages.

**Commands**

* OPENSHUTTER: Opens the dome.
* CLOSESHUTTER: Closes the dome.

**Notes**

* Adjust the parkedThreshold in the isTelescopeParked() function to match your ultrasonic sensor's readings and the telescope's parked position.
* Ensure that the magnetic limit switches are positioned correctly to detect the partial open and close positions.
* Test the dome operation thoroughly after making any changes to the code or hardware.