# functions.py — Hardware and Vision Utilities for Autonomous Robot

This file provides all the core hardware control and image processing functions for your Raspberry Pi robot car.  
It abstracts GPIO operations, PWM setup, DC motor and steering servo commands, and encapsulates reusable vision functions for finding colored contours and displaying debug overlays.

## Main Sections

### 1. GPIO and PWM Setup

* Initializes the Raspberry Pi GPIO in BCM mode.
* Sets up output pins for the L298N motor driver (IN1, IN2, ENA) and the servo (SERVO\_PIN).
* Configures PWM channels for motor speed and servo position.

### 2. Motor Control Functions

#### set\_motor(speed)

* Controls motor direction and speed via L298N.
* speed range: -100 (full reverse) to +100 (full forward), 0 stops.
* Used for every movement or stop command.

#### CENTER\_ANGLE, MIN\_ANGLE, MAX\_ANGLE

* Constants for steering servo position.
* CENTER\_ANGLE is straight ahead, MIN\_ANGLE is full left, MAX\_ANGLE is full right.

#### set\_servo\_angle(angle)

* Sets the servo steering angle, clamped between allowed bounds.
* Converts angle to a PWM duty cycle for accurate positioning.

### 3. High-Level Control Functions

#### write(value)

* Unified interface for controlling either steering or motor speed.
* If value is in [60,160], sets servo; if in [1000,2000], sets motor speed.

#### multi\_write(sequence)

* Executes a sequence of actions: steering angles, speeds, or sleep delays.
* Useful for initialization or executing maneuvers.

#### stop\_car()

* Stops all movement, centers steering, and closes OpenCV windows.
* Called on exit or emergency stop.

#### move\_backward\_with\_mirror(forward\_angle, duration=1, speed=50)

* Moves the car backward, steering mirrored relative to the last forward angle.
* For advanced reversing maneuvers.

### 4. Image Processing Functions

#### display\_roi(img, ROIs, color)

* Draws rectangles (ROIs) on the image for visualizing detection zones.
* Used for debugging and live feedback in OpenCV window.

#### find\_contours(img\_lab, lab\_range, ROI)

* Crops the specified ROI from the LAB image.
* Applies LAB color masking, erodes/dilates to reduce noise.
* Finds and returns contours matching the color mask.

#### max\_contour(contours, ROI)

* Selects the largest contour in a region, computes area and position.
* Returns [maxArea, maxX, maxY, mCnt] for use in steering and wall detection.

#### display\_variables(variables)

* Prints key debug information (areas, angles, detected colors, etc.) to the console.
* Overwrites previous values for real-time updates.

## How This File Is Used

* **Imported in main.py** for all hardware and vision operations.
* **Keeps main loop clean:** Complex hardware logic and image analysis are abstracted away for clarity and reuse.
* **Facilitates debugging:** Functions like display\_roi and display\_variables enable easy real-time monitoring.

## Typical Usage Example

from functions import \*  
  
# Set motor speed and direction  
set\_motor(80) # Move forward  
  
# Set steering angle  
set\_servo\_angle(110) # Straight  
  
# Mask and find contours in a camera image  
contours = find\_contours(img\_lab, rOrange, ROI3)  
area, x, y, cnt = max\_contour(contours, ROI3)  
  
# Stop the car  
stop\_car()

## Troubleshooting

* **Servo or motor not responding:** Check GPIO pin assignment and connections.
* **Vision functions not finding contours:** Ensure correct LAB mask values and ROI placement; tune masks with your LAB mask tuner tool.
* **Debugging:** Use display\_roi and display\_variables to monitor real-time system state.

## References

* [RPi.GPIO documentation](https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/)
* [OpenCV Python reference](https://docs.opencv.org/)
* [L298N Motor Driver](https://www.electronicwings.com/nodemcu/l298n-motor-driver)
* [Servo PWM basics](https://www.raspberrypi-spy.co.uk/2013/07/using-a-servo-with-the-raspberry-pi/)