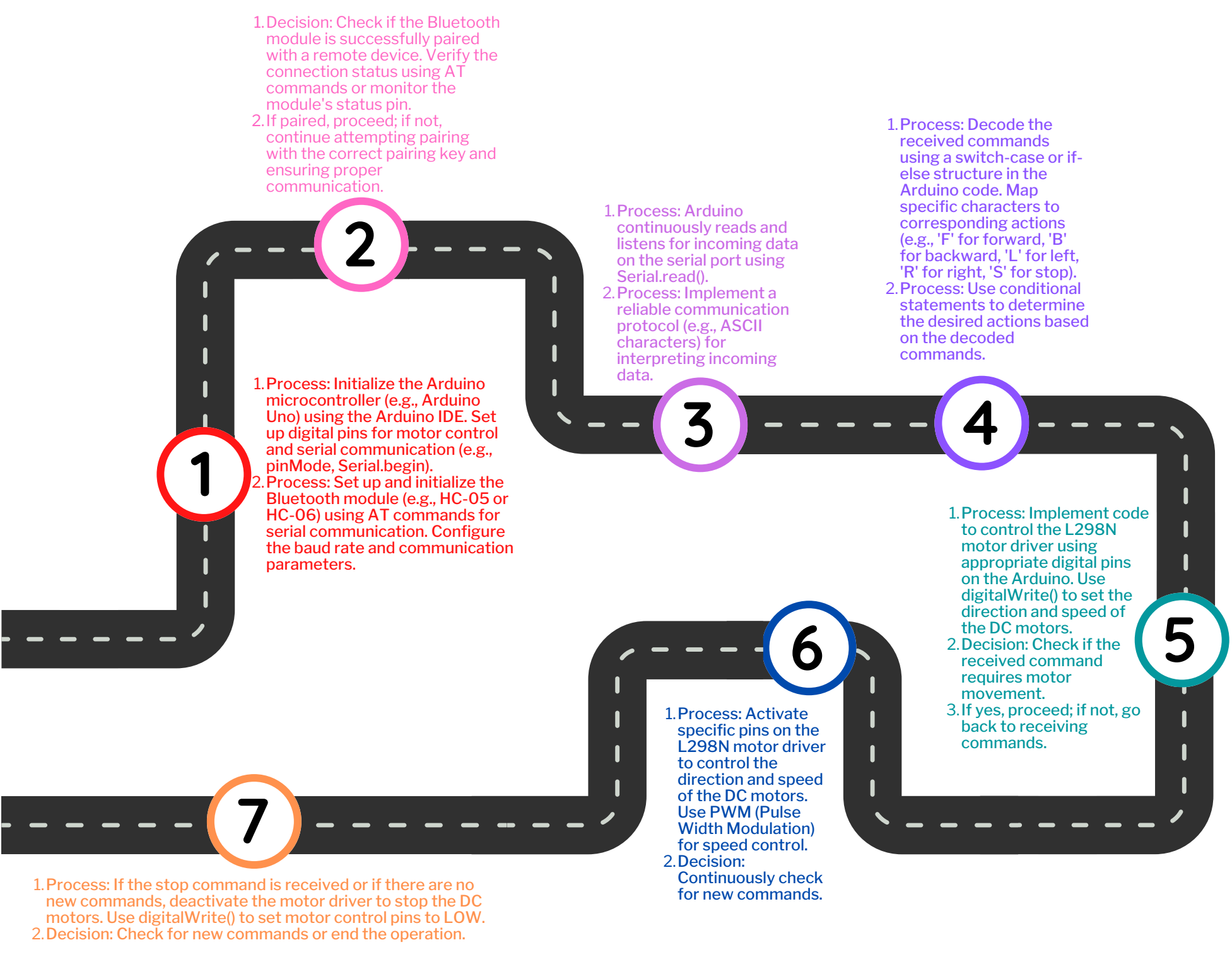
1. Process: Initialize the Arduino microcontroller (e.g., Arduino Uno) using the Arduino IDE. Set up digital pins for motor control and serial communication (e.g., pinMode, Serial.begin).
2. Process: Set up and initialize the Bluetooth module (e.g., HC-05 or HC-06) using AT commands for serial communication. Configure the baud rate and communication parameters.
1
1. Decision: Check if the Bluetooth module is successfully paired with a remote device. Verify the connection status using AT commands or monitor the module's status pin.
2. If paired, proceed; if not, continue attempting pairing with the correct pairing key and ensuring proper communication.
2
1. Process: Arduino continuously reads and listens for incoming data on the serial port using Serial.read().
2. Process: Implement a reliable communication protocol (e.g., ASCII characters) for interpreting incoming data.
3
1. Process: Decode the received commands using a switch-case or if-else structure in the Arduino code. Map specific characters to corresponding actions (e.g., 'F' for forward, 'B' for backward, 'L' for left, 'R' for right, 'S' for stop).
2. Process: Use conditional statements to determine the desired actions based on the decoded commands.
4
1. Process: Implement code to control the L298N motor driver using appropriate digital pins on the Arduino. Use digitalWrite() to set the direction and speed of the DC motors.
2. Decision: Check if the received command requires motor movement.
3. If yes, proceed; if not, go back to receiving commands.
5
1. Process: Activate specific pins on the L298N motor driver to control the direction and speed of the DC motors. Use PWM (Pulse Width Modulation) for speed control.
2. Decision: Continuously check for new commands.
6
1. Process: If the stop command is received or if there are no new commands, deactivate the motor driver to stop the DC motors. Use digitalWrite() to set motor control pins to LOW.
2. Decision: Check for new commands or end the operation.
7