Here's a short documentation for the provided Arduino code, including functions, how it operates, and the connections for connecting Arduino pins to hardware:

Code Documentation

Functions:

1. `setup()`: This function is called once when the Arduino is powered on or reset. It sets up the initial configurations, pin modes, and displays the default RPM and time settings. It also loads the elapsed time from EEPROM for potential resume after a power loss.

2. `loop()`: The main execution loop where the core functionality of the code is implemented. It continuously reads the encoder values, adjusts RPM and time settings, controls motor speed, starts/stops the operation, and toggles between adjusting RPM and time.

3. `EEPROMReadLong(int address)`: Reads a long value from the EEPROM at the specified address.

4. `EEPROMWriteLong(int address, unsigned long value)`: Writes a long value to the EEPROM at the specified address.

Operation:

- The code allows the user to control a motor's speed and operation time. It uses a rotary encoder, two 4-digit 7-segment displays (TM1637), and a push button to achieve this.

- The user can switch between adjusting RPM and time settings using the encoder's push button.

- RPM can be adjusted within the range of 50 to 300 RPM in increments of 10 RPM.

- Time can be set from 0 to 9999 minutes for continuous operation.

- Motor speed is controlled based on an input voltage provided to the voltage control pin.

- The "Start/Stop" button starts and stops the motor operation. The elapsed time is tracked during operation.

- If the time limit is reached, the operation is stopped automatically.

- The code supports a "resume" feature that allows the system to recover from a power loss and continue the operation.

Connections:

- **Display Pin Configuration**:
  - Connect TM1637 displays to Arduino Pins 2 (CLK) and 3 (DIO) for RPM display and Pins 4 (CLK) and 5 (DIO) for time display.

- **Encoder Pin Configuration**:
  - Connect the encoder to Arduino Pins 6 (A channel) and 7 (B channel).

- **Button Pin Configuration**:
  - Use Arduino Pin 8 for the "Start/Stop" button and Pin 11 for the encoder's push button.

- **Motor Control Pin Configuration**:
  - Connect the motor control pin to Arduino Pin 9.

- **Direction Pin Configuration**:
  - Connect the motor direction pin to Arduino Pin 10, configuring it as HIGH or LOW as needed.

- **Voltage Input Pin Configuration**:
  - Use an analog input pin (e.g., A0) to read the input voltage for motor speed control.

Operation:

- Adjust the desired RPM and time using the rotary encoder.
- Press the encoder's push button to toggle between adjusting RPM and time.
- Press the "Start/Stop" button to initiate or stop the motor operation.
- The motor speed adjusts based on the voltage input.
- The displays provide real-time feedback on settings, elapsed time, and remaining time.
- The code supports a "resume" feature to continue the operation after a power loss.

This code is versatile and can be applied to various applications where precise motor control and time management are essential.