Approach :

9 leds per headlights

Certainly! Here's the adapted summary approach for using voltage level shifting to connect the 12V input signals to the Arduino:

1. White LEDs:
   * Use a constant current driver with PWM control for the white LEDs.
   * Connect the VCC (12V) and GND (ground) of the constant current driver to the corresponding power source.
   * Connect the output of the constant current driver to the anodes (positive terminals) of the white LEDs.
   * Connect the cathodes (negative terminals) of the white LEDs to a common ground.
   * Connect the PWM input pin of the constant current driver to a PWM-capable pin on the Arduino.
   * Use PWM signals to control the brightness of the white LEDs collectively.
2. Yellow LEDs:
   * Use a shift register, such as the 74HC595, to individually control the yellow LEDs.
   * Connect the data (SER), clock (SRCLK), and latch (RCLK) pins of the shift register to the corresponding pins on the Arduino.
   * Connect the VCC (5V or 3.3V) and GND (ground) of the shift register to the corresponding power source of the Arduino.
   * Connect the output pins of the shift register to the cathodes (negative terminals) of the yellow LEDs through individual current-limiting resistors.
   * Connect the anodes (positive terminals) of the yellow LEDs to a common positive supply (12V).
3. Input Signals:
   * Connect the DRL (Daytime Running Lights) signal to a voltage divider circuit or logic level shifter to reduce the voltage to a safe level compatible with the Arduino's input pins.
   * Connect the low-beam signal to a voltage divider circuit or logic level shifter to reduce the voltage to a safe level compatible with the Arduino's input pins.
   * Connect the turn signal signal to a voltage divider circuit or logic level shifter to reduce the voltage to a safe level compatible with the Arduino's input pins.
4. Arduino Code:
   * Read the state of the DRL, low-beam, and turn signal input pins using digitalRead() in the Arduino code.
   * Implement the desired logic based on the input signal conditions to control the white and yellow LEDs.
   * Use PWM to adjust the brightness of the white LEDs collectively when the DRL is active.
   * Use the shift register to control the yellow LEDs individually when required (e.g., during turn signal activation).
   * Turn off the white LEDs and the yellow LEDs when no conditions are met.

By using voltage dividers or logic level shifters, you can safely bring down the 12V input signals from the DRL, low-beam, and turn signal to a level compatible with the Arduino's input pins. This ensures the protection of the Arduino while allowing it to detect and respond to the input signals accurately.

Remember to calculate and choose suitable resistor values for the voltage dividers based on the desired voltage division ratio. Consult the datasheets and documentation of the components you use for proper wiring instructions and compatibility.

If you have any further questions or need additional assistance, please feel free to ask!