**MSPM0L1306 Pin Assignment Report**

**1. Introduction**

This document details the pin assignment strategy for the MSPM0L1306 microcontroller interfacing with two DRV8242 gate drivers. The pin selection process prioritizes optimal routing for critical signals (PWM and IPROPI) while ensuring proper functionality for all required interfaces.

The layout, as shown in the provided diagram, places the microcontroller (MSPM0L1306) in the center with gate drivers positioned on both the left and right sides. Additional interface signals are distributed around the perimeter of the board.

**2. Signal Overview**

The MSPM0L1306 microcontroller needs to interface with the following components:

* Two DRV8242 gate drivers (left and right sides)
* SWD debugging interface (left side)
* SPI communication interface (right side)
* LIN communication interface (right side)
* Multiple feedback and monitoring signals (bottom side)

**3. Pin Assignment Tables**

**Table 1: All Possible Pins for Each Signal**

| **Signal Type** | **Function** | **Compatible MCU Pins** |
| --- | --- | --- |
| **PWM** | Gate driver PWM inputs | PA1 (Pin 2), PA7 (Pin 11), PA9 (Pin 13), PA12 (Pin 16), PA13 (Pin 17), PA14 (Pin 18) |
| **NFAULT** | Fault indication from drivers | Any GPIO input: PA0-PA28 (Pins 1-32) |
| **NSLEEP** | Sleep mode control | Any GPIO output: PA0-PA28 (Pins 1-32) |
| **DRVOFF** | Driver disable control | Any GPIO output: PA0-PA28 (Pins 1-32) |
| **EN/IN1** | Enable/direction control | Any GPIO output: PA0-PA28 (Pins 1-32) |
| **IPROPI** | Current sensing (analog) | PA15 (Pin 19), PA18 (Pin 22), PA19 (Pin 23), PA20 (Pin 24), PA21 (Pin 25), PA22 (Pin 26), PA24 (Pin 28), PA25 (Pin 29), PA26 (Pin 30), PA27 (Pin 31) |
| **SWDIO** | Debug data I/O | PA0 (Pin 1), PA27 (Pin 31) |
| **SWD CLK** | Debug clock | PA1 (Pin 2), PA28 (Pin 32) |
| **SPI MOSI** | SPI data output | PA4 (Pin 8), PA5 (Pin 9), PA10 (Pin 14) |
| **SPI CLK** | SPI clock | PA6 (Pin 10), PA11 (Pin 15) |
| **LIN TX** | UART transmit | PA7 (Pin 11), PA8 (Pin 12), PA12 (Pin 16), PA13 (Pin 17) |
| **LIN RX** | UART receive | PA8 (Pin 12), PA9 (Pin 13), PA14 (Pin 18), PA15 (Pin 19) |
| **FB1-FB4** | Analog feedback inputs | PA15 (Pin 19), PA18 (Pin 22), PA19 (Pin 23), PA20 (Pin 24), PA21 (Pin 25), PA22 (Pin 26), PA24 (Pin 28), PA25 (Pin 29), PA26 (Pin 30), PA27 (Pin 31) |
| **VSave** | Analog voltage monitoring | PA15 (Pin 19), PA18 (Pin 22), PA19 (Pin 23), PA20 (Pin 24), PA21 (Pin 25), PA22 (Pin 26), PA24 (Pin 28), PA25 (Pin 29), PA26 (Pin 30), PA27 (Pin 31) |

**Table 2: Optimized Pin Assignments**

| **Signal Type** | **Function** | **Left Driver Pin** | **Right Driver Pin** |
| --- | --- | --- | --- |
| **PWM** | Gate driver PWM inputs | PA1 (Pin 2) | PA9 (Pin 13) |
| **NFAULT** | Fault indication from drivers | PA5 (Pin 9) | PA3 (Pin 7) |
| **NSLEEP** | Sleep mode control | PA6 (Pin 10) | PA20 (Pin 24) |
| **DRVOFF** | Driver disable control | PA7 (Pin 11) | PA19 (Pin 23) |
| **EN/IN1** | Enable/direction control | PA22 (Pin 26) | PA4 (Pin 8) |
| **IPROPI** | Current sensing (analog) | PA27 (Pin 31) | PA15 (Pin 19) |
| **SWDIO** | Debug data I/O | PA27 (Pin 31) | - |
| **SWD CLK** | Debug clock | PA28 (Pin 32) | - |
| **SPI MOSI** | SPI data output | - | PA8 (Pin 12) |
| **SPI CLK** | SPI clock | - | PA6 (Pin 10) |
| **LIN TX** | UART transmit | - | PA13 (Pin 17) |
| **LIN RX** | UART receive | - | PA14 (Pin 18) |
| **FB1** | Analog feedback input | - | PA25 (Pin 29) |
| **FB2** | Analog feedback input | - | PA26 (Pin 30) |
| **FB3** | Analog feedback input | - | PA18 (Pin 22) |
| **FB4** | Analog feedback input | - | PA20 (Pin 24) |
| **VSave** | Analog voltage monitoring | - | PA15 (Pin 19) |

**4. Signal Descriptions**

**4.1 Gate Driver Signals**

1. **PWM (Pulse Width Modulation)**:
   * **Function**: Controls the switching frequency and duty cycle of the gate drivers.
   * **Implementation**: PWM signals require timer peripherals with PWM capability.
   * **Routing Priority**: Highest priority. Routes should be as short as possible with no vias.
2. **IPROPI (Current Proportional Output)**:
   * **Function**: Analog output from the gate driver that's proportional to motor current.
   * **Implementation**: Requires connection to ADC pins on the microcontroller for current monitoring.
   * **Routing Priority**: High priority. Routes should be short with minimal noise exposure.
3. **NFAULT (Fault Indication, Active Low)**:
   * **Function**: Signals fault conditions in the gate driver (overcurrent, overtemperature, undervoltage).
   * **Implementation**: Connected to GPIO inputs, typically with pull-up resistors.
   * **Routing Priority**: Medium priority.
4. **NSLEEP (Sleep Mode, Active Low)**:
   * **Function**: Places the gate driver in low-power sleep mode when pulled low.
   * **Implementation**: Connected to GPIO outputs from the microcontroller.
   * **Routing Priority**: Medium priority.
5. **DRVOFF (Driver Disable)**:
   * **Function**: Disables the gate driver outputs when asserted.
   * **Implementation**: Connected to GPIO outputs from the microcontroller.
   * **Routing Priority**: Medium priority.
6. **EN/IN1 (Enable/Direction Control)**:
   * **Function**: Controls motor direction (forward/reverse) and enables/disables operation.
   * **Implementation**: Connected to GPIO outputs from the microcontroller.
   * **Routing Priority**: Medium priority.

**4.2 Communication Interfaces**

1. **SWD (Serial Wire Debug)**:
   * **SWDIO**: Serial data I/O for programming and debugging.
   * **SWD CLK**: Clock signal for SWD interface.
   * **Implementation**: Connected to dedicated debug pins on the microcontroller.
2. **SPI (Serial Peripheral Interface)**:
   * **SPI MOSI**: Master Out Slave In data line.
   * **SPI CLK**: Clock signal for synchronizing data transfer.
   * **Implementation**: Connected to pins with SPI peripheral capability.
3. **LIN (Local Interconnect Network)**:
   * **LIN TX**: Transmit line for UART-based LIN communication.
   * **LIN RX**: Receive line for UART-based LIN communication.
   * **Implementation**: Connected to pins with UART peripheral capability.

**4.3 Feedback Signals**

1. **FB1-FB4 (Feedback)**:
   * **Function**: Analog inputs for monitoring system voltages and parameters.
   * **Implementation**: Connected to ADC-capable pins on the microcontroller.
2. **VSave (Voltage Save)**:
   * **Function**: Monitors a specific voltage level, possibly for backup or safety purposes.
   * **Implementation**: Connected to an ADC-capable pin on the microcontroller.

**5. Optimization Methodology**

The pin assignment optimization followed these principles:

1. **Priority-Based Assignment**:
   * PWM and IPROPI signals were given highest priority due to their critical nature.
   * Routes for these signals were minimized and kept as direct as possible.
2. **Functional Compatibility**:
   * Pins were selected based on their peripheral capabilities (PWM, ADC, SPI, UART).
   * Digital I/O signals were assigned to appropriate GPIO pins.
3. **Proximity Optimization**:
   * Pins were selected based on their physical proximity to the connected components.
   * Left driver signals were routed to left-side microcontroller pins when possible.
   * Right driver signals were routed to right-side microcontroller pins when possible.
4. **Signal Integrity Considerations**:
   * Analog signals were kept away from high-frequency digital signals where possible.
   * Power and ground connections were optimized for minimum impedance.

**6. Implementation Guidelines**

**6.1 PCB Layout Recommendations**

1. **Critical Signal Routing**:
   * Route PWM and IPROPI signals directly without vias.
   * Keep traces as short as possible.
   * Maintain consistent impedance for these signals.
2. **Analog Signal Considerations**:
   * Keep analog signals (IPROPI, FB1-4, VSave) away from high-frequency digital lines.
   * Use ground planes to shield analog signals from digital noise.
3. **General Routing Guidelines**:
   * Minimize crossovers between signal traces.
   * Use vias judiciously for less critical signals.
   * Maintain proper clearance between traces.

**6.2 Software Configuration**

1. **Peripheral Setup**:
   * Configure timer peripherals for PWM generation.
   * Set up ADC for current sensing and voltage monitoring.
   * Initialize UART for LIN communication.
   * Configure SPI interface according to application requirements.
2. **Pin Multiplexing**:
   * Initialize pins with appropriate multiplexing settings as defined in the MCU datasheet.
   * Configure GPIO direction (input/output) as required.

**7. Conclusion**

This pin assignment strategy optimizes the interface between the MSPM0L1306 microcontroller and the DRV8242 gate drivers, prioritizing critical signals while ensuring all functional requirements are met. The layout minimizes trace lengths for time-critical and noise-sensitive signals while maintaining proper signal integrity.

By following these pin assignments and implementation guidelines, the design should achieve reliable operation with minimal signal integrity issues. The optimization process balanced multiple factors including electrical requirements, physical layout constraints, and performance priorities.