Subsystem Design

Power Sequencer

1 Description

The power sequencing example demonstrates turning on multiple rails from one start-up, at different intervals. This precaution helps prevent damaging devices during start-up that causes power spikes, bus contention, latch-up errors, and other issues. The MSPM0 allows for use of only one timer to set different intervals for each rail.

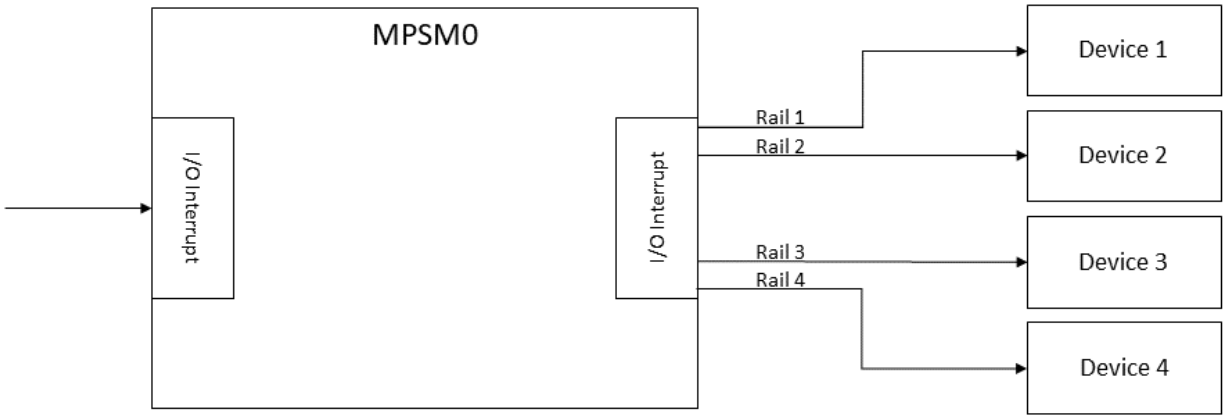

Figure 1-1. Subsystem Functional Block Diagram

2 Required Peripherals

This application requires the one timer, four output pins, and one input pin. The number of output pins can vary depending on application needs.

Table 2-1. Required Peripherals

Subblock Function	Peripheral Use	Notes
Interrupt trigger	1 pin	Signal input for trigger
Output signals	4 pins	Output signals for sequencing
Creating sequence	1 Timer G	Called `TIME_SEQUENCE` in code

3 Compatible Devices

Based on the requirements in Table 2-1, this example is compatible with the devices in Table 3-1. The corresponding EVM can be used for prototyping.

Table 3-1. Compatible Devices

Compatible Devices	EVM
MSPM0Cxxx	LP-MSPM0C1104
MSPM0Lxxx	LP-MSPM0L1306
MSPM0Gxxx	LP-MSPM0G3507

4 Design Steps

1. Determine the desired time interval between each rail at start-up. The time interval set is counted sequentially from rail to rail **not** the start point. See *Design Considerations* for instructions on calculating the intervals.
2. Determine the desired time interval between each rail at shutdown. The time interval set is counted sequentially from rail to rail **not** the start point. Alternatively, outputs can be shutdown all at once.
3. Configure peripherals in SysConfig. Select a timer, configure with a chosen frequency, and set interrupt as a zero event. Set the input interrupt as rising and falling edge. Choose pins for input voltage and output rails.
4. Modify the desired time intervals in the application code. Time intervals are place at the top of the `.c` file.

5 Design Considerations

1. **Multiple rails:** The number of rails for this application can be increased or decreased. Only minor edits are needed to implement the number of rails.
   a. The array size for the interval time sequences needs to match the number of rails chosen. The two arrays referenced are `gTimerUp[]` and `gTimerDown[]`.
   b. If rails are added or subtracted edits need to be made to the `pinToggle` function for each GPIO output.
2. **Sequence order:** The application as written has a specific order for the sequence. To change the order of rails triggered, change the # in GPIO_OUT_PIN_#_PIN that is found in the `pinToggle` function to the desired order in the `if` statement.
3. **Clock settings:** Maximum interval resolution is dependent on the frequency of the timer. Timer clock settings need to be adjusted depending on system clock settings. There is a direct relationship between how fast to clock the timer and the resolution of time between rails. The faster you clock the timer, the more resolution there is between; however, as the input clock frequency increases, the total possible time between rails is decreased.
4. **Calculating intervals:** SysConfig gives period range and resolution based on the set frequency for the MSPM0 family. In the example code, the resolution is 7.81ms while the clock frequency is set to 128Hz. The period for the desired intervals can be calculated by dividing the desired time by the resolution.
5. **Port settings:** Some devices of the MSPM0 family offer multiple ports. If more than one port is in use, the GPIO code portions of the application have to be modified to use multiple ports.
6. **Connection of external devices to output pins:** External devices can be controlled in this type of application in different ways. Three common methods are explored in the following list:
   a. *Enable Pin:* No additional actions are needed for the output.
   b. *Direct Power:* If an external device is being powered by the output, modifications need to be made as well as considerations about the output current limitation located in the device data sheet.
   c. *External Power Circuitry:* If external circuitry is needed for powering of another device, for example an external GPAMP, then the output is like the enable pin situation in 6.a. External circuitry varies on each system, and is beyond the scope of this document.

6 Software Flow Chart

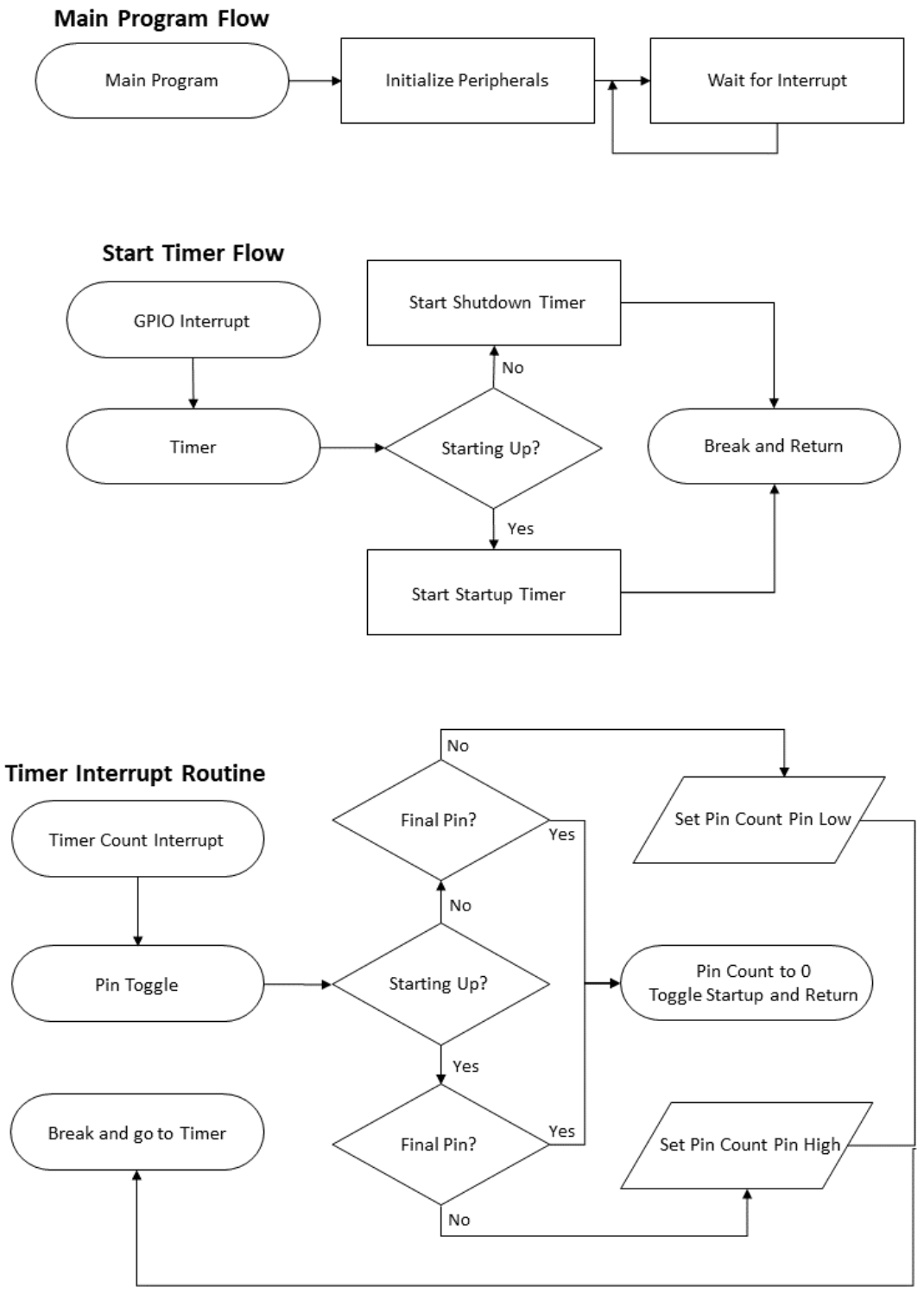

Figure 6-1. Application Software Flow Chart

7 Design Results

Figure 7-1 shows the logic graph results from executing the code example. The end is assuming the pins are also turned off in sequence.

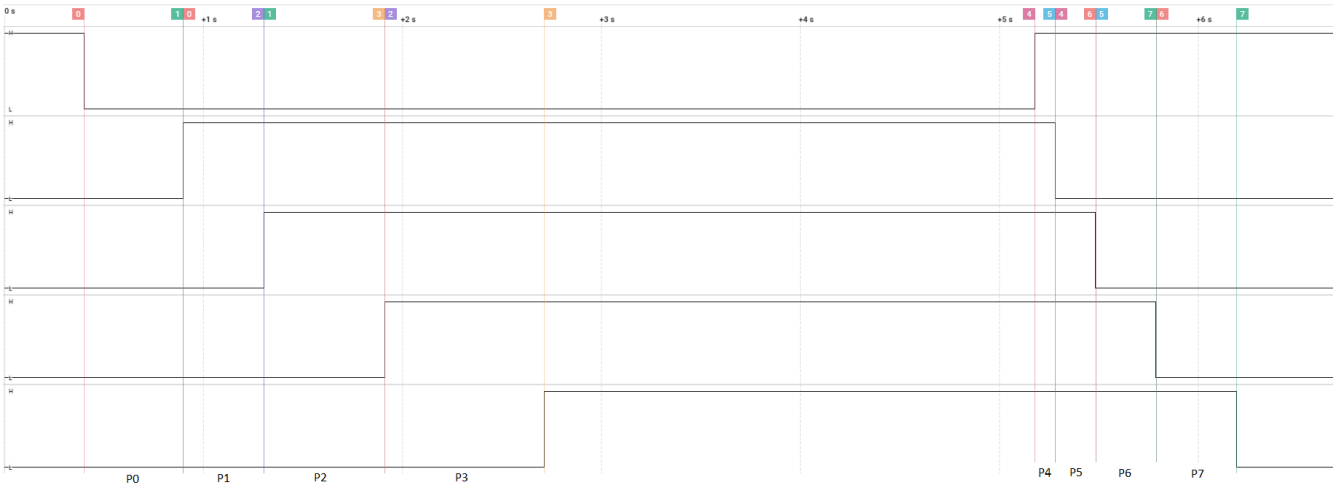

Figure 7-1. Sequence Result Graph

- P0: 498.6ms (2.01Hz)
- P1: 404.72ms (2.47Hz)
- P2: 607.12ms (1.65Hz)
- P3: 801.68ms (1.25Hz)
- P4: 102. 28ms (9.78Hz)
- P5: 202.36ms (4.94Hz)
- P6: 303.56ms (3.29Hz)
- P7: 404.72ms (2.47Hz)

8 Reference

- Texas Instruments, Download the MSPM0 SDK
- Texas Instruments, Learn more about SysConfig
- Texas Instruments, MSPM0L LaunchPad™
- Texas Instruments, MSPM0G LaunchPad™
- Texas Instruments, MSPM0 Timer Academy

9 E2E

See TI's E2E™ support forums to view discussions and post new threads to get technical support for utilizing MSPM0 devices in designs.

10 Trademarks

LaunchPad™ and E2E™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265