PMIC: First one to turn on and the last one to turn off

Embedded Linux Conference 2019

Keerthy Jagadeesh



Agenda

- Introduction to PMIC
- Why do we need PMIC?
- Main components of PMIC
- Regulator driver overview
- Types of regulators
- Other common components
- I2C Debugging in u-boot
- I2C Debugging in kernel
- Hardware debug



Introduction to PMIC

- Power management integrated circuits.
- Communicates with the host Processor via I2C or SPI.
- A PMIC is most commonly used in mobile phones & portable media players to decrease the amount of space required.
- Primary responsibility for implementing power on/off sequence.
- Feeding multiple voltage sources to SoCs
- Power supply types: LDOs, SMPS, DC/DC convertors
- May also provide RTC, GPIO, ADC, Watchdog & even thermal sensor support.
- Some PMICs also include Battery charging support.



Why PMIC

- One stop solution for all the voltage requirements for the whole system including the SoC, RF, external codecs, modem etc.
- Even when multiple regulators are present PMICs offer granular voltage control. Ex: TPS65218.
- Enables deeper power management.
- ADC can be used to monitor variety of voltage and current parameters.
- Thermal sensor can be used to monitor the device temperature.
- RTC Real time clock help keep track of time.



Example: TPS65218



Figure 1. Connection Diagram for TPS65218 and AM335x



Example: TPS659039



DRA7 Power Tree





Types of voltage regulators

• LDO - A low-dropout linear regulator:

- LDOs are simple to regulate, create very little noise
- Requires no inductor for operation
- Produce a highly accurate output voltage
- Use of a pass transistor creates excessive heat, leading to reliability issues and low efficiency levels
- LDOs are most often used in low-power applications where these drawbacks are minimized by the nature of the application

• Switch-mode power supplies (SMPS):

- Most popular power supply today because of their high efficiency
- Store energy in a capacitor or inductor, and repetitively switches its transistor on and off
- Low heat levels & SMPSs typically have efficiency levels above 80%
- High cost and complex design
- SMPSs can create high levels of noise and ripple, which decreases the accuracy of output voltage
- SMPSs are usually used in high power applications



Linux modeling of PMIC

- MFD: It is very natural to model the PMIC as a multi function device. As it caters to Voltage, ADC, GPIO, thermal, RTC etc.
- Most certainly has regulator drivers.
- ADC/GPIO/Thermal/RTC drivers can also be present based on the HW.





MFD Driver nuances

- Initialize the regmap devm_regmap_init_i2c
- Register the interrupt handler & configure the interrupt controller as PMIC can potentially have multiple sources that can generate interrupts.
- Enable probing of child device drivers mfd_add_devices
- Perform any PMIC specific pad configs
- Set any client specific data when multiple versions are supported.



Device Tree Modelling with an example

```
&i2c0 {
    pinctrl-names="default";
    pinctrl-0 = <\&i2c0_pins>;
    clock-frequency = <100000>;
    tps65218: tps65218@24 {
         reg = <0x24>;
         compatible = "ti,tps65218";
         interrupts = <GIC_SPI7 IRQ_TYPE_LEVEL_HIGH>; /* NMIn */
         interrupt-controller;
         #interrupt-cells = <2>;
         dcdc1: regulator-dcdc1 {
                  regulator-name = "vdd_core";
                  regulator-min-microvolt=<912000>;
                  regulator-max-microvolt = <1144000>;
                  regulator-boot-on;
                  regulator-always-on;
         };
```

};

🔱 Texas Instruments

Regulator Driver nuances

- devm_regulator_register: Register all the regulators to the regulator framework.
- Populate necessary hooks for enable/disable. This is turn on/off the regulators.
- Populate get/set voltage hooks to read and write the voltage values.
- One can also set/get current limit hooks.
- Make use of regulator_linear_range to provide vsel to voltage conversion and vice versa.
- map_voltage to map the voltages
- set_ramp_delay : To take care of the slew rate that allows regulators ramp up and ramp down time.



Types of regulators in Linux

- Fixed regulators: Voltage is fixed. Only on/off support.
- Variable voltage DC-DC convertors, SMPS, LDO regulators.
- Variable regulators cater to a range of voltages with fixed number of steps.
- Allows DVFS dynamic voltage frequency scaling.
- Allows to lower voltages during low power modes.
- Variable voltage regulators are also used as cooling agents.
- GPIO Regulators: GPIOs can also be modelled as regulators where in the high value corresponds to a discrete value of voltage and the low corresponds to another discrete value.



Regulator Device Tree nodes

```
dcdc1: regulator-dcdc1 {
    regulator-name = "vdd_core";
    regulator-min-microvolt = <912000>;
    regulator-max-microvolt = <1144000>;
    regulator-boot-on;
    regulator-always-on;
```

```
};
```

};

```
ldo1: regulator-ldo1 {
	regulator-min-microvolt = <180000>;
	regulator-max-microvolt = <180000>;
	regulator-boot-on;
	regulator-always-on;
```

🔱 Texas Instruments

Regulator Device Tree nodes - 2

};



GPIO Driver nuances

- devm_gpiochip_add_data: One time registration with the gpio framework
- Gpio set/get direction to either configure as input or output.
- GPIO set get to retrieve the value of gpio level.
- Set_config enables more config options.
- Gpio_request: optional hook for chip-specific activation, such as enabling module power and clock; may sleep
- Gpio_free: optional hook for chip-specific deactivation, such as disabling module power and clock; may sleep
- Set_multiple: Assigns output values for multiple signals



TPS659038 GPIO Device Tree

```
• &i2c1 {
```

};

```
status = "okay";
clock-frequency = <400000>;
```

```
tps659038:tps659038@58 {
    compatible = "ti,tps659038";
    reg = <0x58>;
    interrupt-parent = <&gpio1>;
    interrupts= <0 IRQ_TYPE_LEVEL_LOW>;
    #interrupt-cells= <2>;
    interrupt-controller;
    ti,system-power-controller;
    ti,palmas-override-powerhold;
    tps659038_gpio:tps659038_gpio {
        compatible = "ti,palmas-gpio";
        gpio-controller;
        #gpio-cells= <2>;
    };
```

```
gpio_fan: gpio_fan {
    /* Based on 5v 500mA AFB02505HHB */
    compatible = "gpio-fan";
    gpios= <&tps659038_gpio2 GPIO_ACTIVE_HIGH>;
    gpio-fan,speed-map = <0 0>, <13000 1>;
    #cooling-cells= <2>;
};
```



RTC Driver nuances

- Clear all the prior rtc interrupts
- Populate the rtc dev ops
 - .read_time
 - .set_time
 - .read_alarm
 - .set_alarm
 - .alarm_irq_enable
- devm_rtc_device_register
- Register the rtc interrupt using devm_request_threaded_irq



TPS659038 RTC Device Tree

```
• &i2c1 {
```

status = "okay"; clock-frequency = <400000>;

```
tps659038: tps659038@58 {
    compatible = "ti,tps659038";
    reg = <0x58>;
    interrupt-parent = <&gpio1>;
    interrupts = <0 IRQ_TYPE_LEVEL_LOW>;
```

#interrupt-cells = <2>;
interrupt-controller;

```
ti,system-power-controller;
ti,palmas-override-powerhold;
```

```
tps659038_rtc:tps659038_rtc {
    compatible = "ti,palmas-rtc";
    interrupt-parent = <&tps659038>;
    interrupts = <8 IRQ_TYPE_EDGE_FALLING>;
    wakeup-source;
};
```

};



Debugging

- U-boot & kernel prints to start with to trace the i2c register values
- Cat /sys/class/regulator/regulator*/*
 - Microvolts gives the actual voltage value
 - State tells whether regulator is turned on or not
 - Microamps gives current drawn
 - Other sysfs nodes that give out a lot of information about regulator.
- Enable CONFIG_CMD_REGULATOR in u-boot to get the regulator state and voltage.



U-boot I2C debug tools

- First need to assign i2c device using i2c dev 0
- One can dump the i2c registers of a particular Slave ID using: i2c md 'slave_id'
- One can modify the registers using any of the i2c mw or i2c mm
- Example: i2c dev 0 Setting bus to 0

i2c md 0x58 0x20

0020: 11 00 be 3a 00 00 be 3e 00 00 00 00 11 00 c7 32 ...:..>.....2 i2c mw 0x58 0x23 0x3B

i2c md 0x58 0x20 0020: 11 00 be 3b 00 00 be 3e 00 00 00 00 11 00 c7 32 ...;..>.....2 i2c speed

Current bus speed=400000



Kernel I2C debug tools

- i2cget [-f] [-y] I2CBUS CHIP-ADDRESS [DATA-ADDRESS [MODE]]

Command: i2cget -f -y 0x0 0x58 0x23 Output: 0x3b



Kernel I2C debug tools - 2

 i2cset [-f] [-y] [-m MASK] [-r] I2CBUS CHIP-ADDRESS DATA-ADDRESS [VALUE] ... [MODE]



I2C Protocol fundamentals



- Start: the master device leaves SCL high and pulls SDA low
- Address Frame: 7-bit address
- R/W: a R/W bit indicating whether this is a read (1) or write (0) operation
- NACK/ACK bit: receiving device pulls the SDA line low before the 9th clock pulse means ACK otherwise it is a NACK.
- Data Frames: The master will simply continue generating clock pulses at a regular interval, and the data will be placed on SDA by either the master or the slave, depending on whether the R/W bit indicated a read or write operation
- Stop: Stop conditions are defined by a 0->1 (low to high) transition on SDA after a 0->1 transition on SCL, with SCL remaining high



I2C Hardware debug - 2

Example showing a normal i2c write sequence followed by Ack





I2C Hardware debug

Probe the I2C Clock and Data lines

Check with the standard expected patterns for read/write

Any discrepancy means HW is not behaving as expected

For example Read Nack





I2C Hardware debug - 3

Disc. Disc. Color Color 1051941 1020.862.72 80.5 us 18 S 56 Wither Transaction 4** 4** 50thare Capture Buffer 1051942 1020.862.827 80.5 us 18 S 56 Wither Transaction 4**	Index	mis me lis	Dur	Len E	rr S/P	Addr	Record	Data	^	Capture Control	
1001 1122082.78 0.0 so 100 0.0 so 100 0.0 so	1001941	10-20 962 749	50 G U0	1 D	en	E0	Read Transaction	L'ata		Coffman Castor Differ	
100141 1022062.00 00 0	1691842	10:20.862.827	80.5 us	1.8	5	58	Write Transaction	10		Software Capture Burler	
109 102 0 000 000 0000 000 000 000 000	1691843	10:20.862.908	56.6 us	1.8	SP	58	Read Transaction	45*			1111111
1691845 10.20.865.950 85.5 is 18 S 58 Write Transaction 23 1691847 10.20.867.036 65.3 us 18 S 58 Write Transaction 20 1691847 10.20.867.036 65.3 us 18 S 58 Write Transaction 21 1691844 10.20.867.199 65.1 us 18 S 58 Write Transaction 23 1691840 10.20.867.359 65.6 us 18 S 88 Write Transaction 37* 1691851 10.20.867.578 80.7 us 18 S 88 Write Transaction 37* 1691852 10.20.867.578 80.7 us 18 S 58 Write Transaction 23 1691855 10.20.867.578 80.5 us 8 Write Transaction 23 7* bit Slave Device (50 (B 0 1691855 10.20.871.104 85.6 us 18 S 8 Write Transaction 23 7* bit Slave Device (20 (B 0 1691855 10.20.871.104 85.6 us 18 S 8 Write Transaction	1691844	10:20 863 000	81 7 us	2 B	SP	58	Write Transaction	23.37			0-1
1691846 10 20.867.036 56 3 us 1 B SP 58 Read Transaction 37* 1691847 10 20.867.117 61 8 us 1 B SP 58 Read Transaction 20 1691844 10 20.867.137 80 6 us 1 B SP 58 Read Transaction 23 1691845 10 20.867.338 65 3 us 1 B SP 58 Write Transaction 23 1691845 10 20.867.438 60 5 us 1 B SP 58 Write Transaction 23 1691852 10 20.867.786 66 5 us 1 B SP 58 Write Transaction 23 1691854 10 20.867.787 66 5 us 1 B SP 58 Write Transaction 23 0 0 7-bit Slave Device (50 (B 0 0 7-bit Slave Device (50 (B 0 0 7-bit Slave Device (50 (B 0 0 0 0 7-bit Slave Device (50 (B 0 0 0 0 7-bit Slave Device (50 (B 0 0 0 0 7-bit Slave Device (50 (B 0 0 0 7-bit Slave Dev	1691845	10:20 866 950	85.6 us	1 B	S	58	Vite Transaction	23			0
199147 1020.867.117 81.8 us 18 \$ 58 Write Transaction 20 1991444 1020.867.178 0.6 us 18 \$ 58 Write Transaction 23 Description Tors By 1691845 1020.867.738 56.3 us 18 \$ 58 Write Transaction 37* Description Tors By 1691850 1020.867.738 56.3 us 18 \$ 58 Write Transaction 37* Description Tors By 1691854 1020.867.758 66.1 us 18 \$ 58 Read Transaction 37* Description Tors By 1691854 1020.867.678 66.6 us 18 \$ 58 Read Transaction 23 Description Tors By 1691855 1020.867.678 66.6 us 18 \$ 58 Write Transaction 23 Description Tors By Description	1691846	10:20.867.036	56.3 us	1 B	SP	58	Read Transaction	37*		Maránakas	
199184a 10.20.867 199 56.1 us 1 S 58 Read Transaction 11 * Description Tors B) 1691844 10.20.867 328 0.60 us 1 S 58 • Read Transaction 23 • It is the vector (S0 (B 169)18 • It is the vector (S0 (B 16	1691847	10:20.867.117	81.8 us	1 B	S	58	Vite Transaction	20		Navigator	
1691449 1020.867.278 80.6 us 1 8 \$ 58 NMme Transaction 23 1691850 1020.867.359 65.3 us 1 8 S 68 NMme Transaction 37* 1691851 1020.867.378 65.1 us 1 8 S 68 NMme Transaction 37* 1691852 1020.867.578 65.1 us 1 8 S 58 NMme Transaction 37* 1691854 1020.867.578 65.6 us 1 8 S 58 NMme Transaction 23 1691855 1020.867.578 65.6 us 1 8 S 58 NMme Transaction 23 - <t< td=""><td>1691848</td><td>10:20.867.199</td><td>56.1 us</td><td>1 B</td><td>SP</td><td>58</td><td>Read Transaction</td><td>11*</td><td></td><td>Description Txn</td><td>.s Bytes</td></t<>	1691848	10:20.867.199	56.1 us	1 B	SP	58	Read Transaction	11*		Description Txn	.s Bytes
1691850 10 20.867,389 66 3 us 1 B SP 68 Read Transaction 37* 1691851 10 20.867,389 66 3 us 1 B SP 68 Read Transaction 23 1691852 10 20.867,587 80 7 us 1 B SP 68 Read Transaction 37* 1691853 10 20.867,578 66 6 us 1 B SP 68 Read Transaction 37* 1691854 10 20.867,678 66 6 us 1 B SP 68 NMm Transaction 23 0 1691855 10 20.877,764 61 6 us 2 B SP 68 NMm Transaction 23 0 0 7* bit Slave Device (20) (B 0 1691855 10 20.877,764 65 6 us 1 B SP 58 NMm Transaction 23 0 0 7* bit Slave Device (20) (B 0 0 0* 7* bit Slave Device (20) (B 0 0 0* 7* bit Slave Device (20) (B 0 0* 0* 7* bit Slave Device (20) (B 0 0* 0* 7* bit Slave Device (20) (B 0 0* 0* 7* bit Slave Device (20) (B 0 0*	1691849	10:20.867.278	80.6 us	1 B	S	58	Vite Transaction	23		₩ 12C Bus	
1691651 102.087.488 80.5 us 18 S 58 Write Transaction 23 1691652 102.087.518 56.1 us 18 S 58 Nead Transaction 37 Inchaoma Slave Device Up() 0 0 7-bit Slave Device (50) (8 0 0 0 0 7-bit Slave Device (50) (8 0 <	1691850	10:20.867.359	56.3 us	1 B	SP	58	Read Transaction	37*		7 - bit Slave Device (59) /P 160	1 19/56
1691652 10 20 867 518 66 1 us 18 SP 58 Read Transaction 37* 1691653 10 20 867 577 60 7 us 18 S 58 Write Transaction 37* 1691654 10 20 867 767 66 6 us 18 SP 58 Read Transaction 37* 1691654 10 20 867 7674 81 6 us 28 SP 58 Read Transaction 23 0 0 7* bit Slave Device (21) (8 0 1691656 10 20 877 764 81 6 us 18 SP 58 Read Transaction 23 0	1691851	10:20.867.438	80.5 us	1 B	S	58	Write Transaction	23		P II I CL D	0
1691653 10 20.867.678 66 us 1 B S 58 N/His Transaction 23 1691654 10 20.867.678 56 6 us 1 B S P 58 Read Transaction 23 40 1691655 10 20.877.64 81 6 us 2 B S P 58 Write Transaction 23 40 1691656 10 20.877.64 81 6 us 2 B S P 58 Write Transaction 23 40 1691656 10 20.870.648 81 0 us 1 B S 58 Write Transaction 23 1691656 10 20.871.027 77 9 us 1 B S 58 Write Transaction 20 1691650 10 20.871.104 56 6 us 1 B S P 58 Read Transaction 23 1691651 10 20.871.344 77 9 us 1 B S 58 Write Transaction 23 1691651 10 20.871.344 77 9 us 1 B S 58 Write Transaction 23 1691656 10 20.871.427 56 5 us 1 B S 58 Write Transaction 23 1691656 10 20.871.427 65 6 us 1 B S 58 Write Transaction 23 <td>1691852</td> <td>10:20.867.518</td> <td>56.1 us</td> <td>1 B</td> <td>SP</td> <td>58</td> <td>Read Transaction</td> <td>37*</td> <td></td> <td>Unknown Slave Device</td> <td>U</td>	1691852	10:20.867.518	56.1 us	1 B	SP	58	Read Transaction	37*		Unknown Slave Device	U
1691854 1020.867.778 56 6 us 1 B SP 58 Read Transaction 37* 1691855 1020.867.764 61 6 us 2 B SP 58 Write Transaction 23 40 1691855 1020.877.054 61 6 us 1 B S 58 Write Transaction 23 40 1691856 1020.870.0568 81 0 us 1 B S S8 Write Transaction 23 40 1691857 1020.870.0349 56 6 us 1 B S S8 Write Transaction 20 1691858 1020.871.014 66 6 us 1 B S S8 Read Transaction 11* 1691861 1020.871.179 77.4 us 1 B S S8 Write Transaction 23 1691862 1020.871.424 77.9 us 1 B S S8 Write Transaction 23 Statistics Exameration 1691862 1020.871.424 77.9 us 1 B S S8 Write Transaction 23 Cick on the bus or a device to see statistics. Cick on the bus or a device	1691853	10:20.867.597	80.7 us	1 B	S	58	Nrite Transaction	23		> # 7-bit Slave Device (50) (B	0
1691865 10 20.877.764 81 6 us 2 8 \$P 58 Write Transaction 23 40 1691855 10 20.870.868 81 0 us 1 B \$P 58 Write Transaction 23 1691855 10 20.870.868 61 us 1 B \$P 58 Pead Transaction 40* 1691856 10 20.871.027 77.9 us 1 B \$P 58 Pead Transaction 10* 1691865 10 20.871.107 77.9 us 1 B \$P 58 Pead Transaction 23 1691861 10 20.871.179 77.4 us 1 B \$P 58 Pead Transaction 23 1691861 10 20.871.479 77.4 us 1 B \$P 58 Pead Transaction 23 1691861 10 20.871.484 77.9 us 1 B \$P 58 Write Transaction 23 1691864 10 20.871.487 7.9 us 1 B \$P 58 Write Transaction 23 Click on the bus or a device to see statistics. 1691865 10 20.871.487 65 cus 1 B \$P \$P Write Transaction	1691854	10:20.867.678	56.6 us	1 B	SP	58	Read Transaction	37*		From the second seco	0
1691855 1020.870.868 81.0 us 18 S 58 NMme Transaction 23 1691857 1020.870.949 56.6 us 18 S 58 Nead Transaction 20 1691858 1020.871.104 56.6 us 18 SP 58 Note Transaction 20 1691858 1020.871.104 56.6 us 18 SP 58 Note Transaction 20 1691850 1020.871.104 56.6 us 18 SP 58 Note Transaction 23 1691851 1020.871.257 56.5 us 18 SP 58 Read Transaction 40* 1691852 1020.871.427 79.9 us 18 S 58 Note Transaction 23 1691853 1020.871.427 75.6 us 18 SP 58 Read Transaction 40* 1691854 1020.871.427 75.6 us 18 SP 58 Nead Transaction 23 1691856 1020.871.472 78.0 us 18 SP 58 Nead Transaction 23 Click on the bus or a device to see statistise.	1691855	10:20.867.764	81.6 us	2 B	SP	58	Nrite Transaction	23 40		From the second seco	0
1691657 10 20 870 349 56 6 us 1 B SP 58 • Read Transaction 20 1691658 10 20 871 027 77 9 us 1 B S 58 • Write Transaction 20 1691659 10 20 871 104 66 us 1 B SP 58 • Read Transaction 11 * 1691660 10 20 871 179 77 4 us 1 B S 58 • Write Transaction 23 1691661 10 20 871 444 77 9 us 1 B S 58 • Write Transaction 40 * 1691661 10 20 871 422 56 S us 1 B S 58 • Write Transaction 23 1691663 10 20 871 422 56 S us 1 B S 58 • Write Transaction 23 1691665 10 20 871 457 56 S us 1 B S 58 • Write Transaction 23 Click on the bus or a device to see statistics. 1691665 10 20 871 467 62 4 us 1 B S 58 Write Transaction 23 Click on the bus or a device to see statistics. Statistic Visible Au 1691667 10 20	1691856	10:20.870.868	81.0 us	1 B	S	58	Nrite Transaction	23		From the second seco	0
1691686 10 20.871 027 77.9 us 18 S 58 Write Transaction 20 1691650 10 20.871 104 65 6 us 18 S 68 Read Transaction 11* 1691650 10 20.871 179 77.4 us 18 S 58 Write Transaction 23 1691650 10 20.871 127 56.5 us 18 SP 58 Write Transaction 23 1691650 10 20.871 422 56.5 us 18 SP 58 Pread Transaction 23 1691650 10 20.871 422 56.5 us 18 SP 58 Pread Transaction 23 1691650 10 20.871 422 56.5 us 18 SP 58 Read Transaction 23 1691656 10 20.871 461 81 9 us 28 SP 58 Write Transaction 23 Cick on the bus or a device to see statistics. 1691656 10 20.874 4549 65 cu us 18 S 58 Write Transaction 23 Cick on the bus or a device to see statistics. Statistic Visible Av 1691657 10 20.874 4579 <td< td=""><td>1691857</td><td>10:20.870.949</td><td>56.6 us</td><td>1 B</td><td>SP</td><td>58</td><td>Read Transaction</td><td>40*</td><td></td><td>> # 7-bit Slave Device (40) (B</td><td>0</td></td<>	1691857	10:20.870.949	56.6 us	1 B	SP	58	Read Transaction	40*		> # 7-bit Slave Device (40) (B	0
1691650 1020.871.104 56.6 us 1 B SP 58 New Fransaction 23 1691650 1020.871.179 77.4 us 1 B S 58 NVine Transaction 23 1691650 1020.871.1257 56.5 us 1 B SP 58 Read Transaction 40* 1691651 1020.871.424 77.9 us 1 B S 58 Note Transaction 23 1691652 1020.871.424 77.9 us 1 B S 58 Read Transaction 23 1691656 1020.871.424 77.6 us 1 B S 58 New Fransaction 23 1691656 1020.871.676 65 us 1 B SP 58 New Fransaction 23 1691656 1020.871.429 85 2 us 1 B SP 58 New Fransaction 23 Click on the bus or a device to see statistics. 1691657 1020.874.429 85 2 us 1 B S 58 NVine Transaction 23 Click on the bus or a device to see statistics. 1691670 1020.874.429 81 us 1 B S 58 <td>1691858</td> <td>10:20.871.027</td> <td>77.9 us</td> <td>1 B</td> <td>S</td> <td>58</td> <td>Write Transaction</td> <td>20</td> <td></td> <td>F T bit Slave Device (40) (5</td> <td>ů.</td>	1691858	10:20.871.027	77.9 us	1 B	S	58	Write Transaction	20		F T bit Slave Device (40) (5	ů.
1691860 10 20 871.179 77.4 us 18 S 58 • Mine Transaction 23 1691861 10 20 871.371 56 5 us 18 S 58 • Read Transaction 23 1691862 10 20 871.422 56 5 us 18 S 58 • Read Transaction 23 1691863 10 20 871.422 56 5 us 18 S 58 • Write Transaction 23 1691864 10 20 871.476 56 5 us 18 S 58 • Write Transaction 23 1691865 10 20 871.476 56 5 us 18 S 58 • Read Transaction 23 • Clk con the bus or a device to see statistics. 1691865 10 20 871.476 56 5 us 18 S 58 • Write Transaction 23 • Clk con the bus or a device to see statistics. Statistic Visible Av 1691866 10 20 871.459 56 1 us 18 S 58 • Write Transaction 20 • Statistic Visible Av 1691870 10 20 874.467 66 rus 18 S 58 • Read Transaction 11 • Is <td>1691859</td> <td>10:20.871.104</td> <td>56.6 us</td> <td>1 B</td> <td>SP</td> <td>58</td> <td>Read Transaction</td> <td>11*</td> <td></td> <td>7 T-Dit Slave Device (OE) (B</td> <td>0</td>	1691859	10:20.871.104	56.6 us	1 B	SP	58	Read Transaction	11*		7 T-Dit Slave Device (OE) (B	0
1691661 10 20.871.327 56.5 us 1 B S P 58 Read Transaction 2.3 1691652 10 20.871.344 77.9 us 1 B S 58 N/He Transaction 2.3 Image: Control of the standard s	1691860	10:20.871.179	77.4 us	1 B	S	58	Write Transaction	23		Chalictics Enumeration	
1691662 10 20.871.434 77.9 us 1B S 58 NMme Transaction 23 1691663 10 20.871.426 78 0 us 1B S 58 Pead Transaction 23 1691663 10 20.871.476 56 5 us 1B SP 58 Write Transaction 23 Click on the bus or a device to see statistics. 1691665 10 20.871.476 56 5 us 1B SP 58 Write Transaction 23 Click on the bus or a device to see statistics. Statistic Visible Av 1691665 10 20.871.429 85 2 us 1B S 58 Write Transaction 23 Click on the bus or a device to see statistics. Statistic Visible Av 1691667 10 20.874.429 85 2 us 1B S 58 Write Transaction 20 Statistic Visible Av 1691670 10 20.874.629 81 G us 1B S 58 Write Transaction 23 Visible Av 1691671 10 20.874.629 81 G us 1B S S Write Transaction 23 Visible Av 1691671 10 20.874.629 </td <td>1691861</td> <td>10:20.871.257</td> <td>56.5 us</td> <td>1 B</td> <td>SP</td> <td>58</td> <td>Read Transaction</td> <td>40*</td> <td></td> <td>Stausucs</td> <td></td>	1691861	10:20.871.257	56.5 us	1 B	SP	58	Read Transaction	40*		Stausucs	
1691663 10 20.871.422 66.5 us 1 B S P 58 Read Transaction 2.3 1691654 10 20.871.498 78.0 us 1 B S 68 Write Transaction 2.3 Clock on the bus or a device to see statistics. 1691656 10 20.871.476 66.5 us 1 B S P 58 Read Transaction 2.3 Clock on the bus or a device to see statistics. 1691656 10 20.871.498 78.0 us 1 B S P 58 Write Transaction 2.3 Clock on the bus or a device to see statistics. 1691656 10 20.871.498 66.6 us 1 B S P 58 Write Transaction 2.3 Clock on the bus or a device to see statistics. 1691656 10 20.874.499 65.6 us 1 B S P 58 Note Transaction 42* 1691650 10 20.874.629 81.6 us 1 B S 58 Write Transaction 2.3 1691871 10 20.874.629 81.6 us 1 B S 58 Write Transaction 2.3 1691872 10 20.874.629 81.6 us 1 B S 58 New Transaction 2.3 1691871	1691862	10:20.871.344	77.9 us	1 B	S	58	Write Transaction	23		Dravious Next	
1691664 1020.871.488 78.0 us 1 B S 58 • Write Transaction 23 1691665 1020.871.676 65 us 1 B S P 58 • Pread Transaction 23 4F 1691666 1020.871.676 61 us 2 B S P 58 Write Transaction 23 4F 1691666 1020.871.676 61 us 2 B S P 58 Write Transaction 23 4F 1691667 1020.874.384 65 G us 1 B S P 58 • Read Transaction 20 1691670 1020.874.484 66 G us 1 B S P 58 • Read Transaction 20 1691671 1020.874.659 81 E us 1 B S P 58 • Read Transaction 20 1691671 1020.874.659 81 E us 1 B S P 58 • Read Transaction 23 1691671 1020.874.659 81 E us 1 B S P 68' • Read Transaction 24 1691671 1020.874.679 81 E us 0 B S P 58' • Read Transaction 24 1691671 1020.874.679 81 E us 0 B S P 58' • Read Transaction <td>1691863</td> <td>10:20.871.422</td> <td>56.5 us</td> <td>1 B</td> <td>SP</td> <td>58</td> <td>Read Transaction</td> <td>40*</td> <td></td> <td>TICYIOUS INCAC O</td> <td></td>	1691863	10:20.871.422	56.5 us	1 B	SP	58	Read Transaction	40*		TICYIOUS INCAC O	
1691865 102.0871.576 56.5 us 1 B SP 58 Note Transaction 2.0 1691866 102.0871.671 65.6 us 1 B S 58 Note Transaction 2.3 47 1691866 102.0874.299 85.2 us 1 B S 58 Note Transaction 2.3 1691866 102.0874.429 85.4 us 1 B S 58 Winte Transaction 2.3 1691866 102.0874.459 56.7 us 1 B S 58 Winte Transaction 2.0 1691870 102.0874.459 56.7 us 1 B S 58 Net Transaction 2.3 1691872 102.0874.479 56.7 us 1 B S 58 Net Transaction 2.3 1691872 102.0874.629 81 B S 58 Net Transaction 2.3 Visible Au Visible Note 1 B S 58 Net Transaction 2.3 Visible Au Visible Note 1 B S 58 Note 1 Transaction 2.3 Visible Au Visible Note 1 B S <td>1691864</td> <td>10:20.871.498</td> <td>78.0 us</td> <td>1 B</td> <td>S</td> <td>58</td> <td>Write Transaction</td> <td>23</td> <td></td> <td>Click on the bus or a device to see sta</td> <td>tistics.</td>	1691864	10:20.871.498	78.0 us	1 B	S	58	Write Transaction	23		Click on the bus or a device to see sta	tistics.
1691666 10/20.871.661 81.9 us 2 B SP 58 Write Transaction 2.3 47 1691667 10/20.874.399 65 Lus 1 B S 58 Write Transaction 2.3 1691666 10/20.874.384 65 G tus 1 B S 58 Write Transaction 47* 1691669 10/20.874.467 82.4 us 1 B S 58 Read Transaction 14* 1691870 10/20.874.467 82.4 us 1 B S 58 Write Transaction 2.0 1691870 10/20.874.629 81.6 us 1 B S 58 Write Transaction 2.0 1691871 10/20.874.629 81.6 us 1 B S 58 Write Transaction 2.3 1691872 10/20.874.629 81.6 us 1 B S 58 Write Transaction 2.3 1691872 10/20.874.629 81.6 us 8 SP 68* Read Transaction 2.3 1691872 10/20.874.629 81.6 us 1 B S 58 Write Transaction 2.3 1691874	1691865	10:20.871.576	56.5 us	1 B	SP	58	Read Transaction	40*			
1691667 1020.874.299 85.2 us 1 B S 58 New Transaction 23 1691686 1020.874.384 56 G us 1 B S 58 New Transaction 20 1691696 1020.874.467 62.4 us 1 B S 58 New Transaction 20 1691670 1020.874.459 56.7 us 1 B S 58 Ninte Transaction 20 1691671 1020.874.549 56.7 us 1 B S 58 Ninte Transaction 23 1691671 1020.874.529 81 G us 1 B S 58 Ninte Transaction 23 1691672 1020.874.710 29.1 us 0 B SP 58* Read Transaction 23 1691671 1020.874.710 29.1 us 0 B SP 58* Read Transaction 23 1691674 1020.874.710 29.1 us 0 B SP 58* New Transaction 23 1691674 1020.874.710 29.1 us 0 B SP 58* New Transaction 10 1691674 1020.874.974 <td>1691866</td> <td>10:20.871.661</td> <td>81.9 us</td> <td>2 B</td> <td>SP</td> <td>58</td> <td>Write Transaction</td> <td>23 4F</td> <td></td> <td>Statistic Visit</td> <td>ie Availa</td>	1691866	10:20.871.661	81.9 us	2 B	SP	58	Write Transaction	23 4F		Statistic Visit	ie Availa
1691668 10 20.874.384 56 6 us 1 B S P 58 Read Transaction 20 1691679 10 20.874.467 82.4 us 1 B S 58 Write Transaction 20 1691670 10 20.874.649 56.7 us 1 B S P 58 Read Transaction 11 1691671 10 20.874.629 81.6 us 1 B S P 58 Read Transaction 23 1691872 10 20.874.710 29.1 us 0 B S P 58 Read Transaction 23 C Colspan="4">Colspan="4">Colspan="4">Colspan="4">Colspan="4">Colspan="4">Colspan="4">Colspan="4">Colspan="4" Colspan="4">Colspan="4" Colspan="4" Colspan="4" <td colspan="</td> <td>1691867</td> <td>10:20.874.299</td> <td>85.2 us</td> <td>1 B</td> <td>S</td> <td>58</td> <td>Write Transaction</td> <td>23</td> <td></td> <td></td> <td></td>	1691867	10:20.874.299	85.2 us	1 B	S	58	Write Transaction	23			
1691669 1020.874.467 82.4 us 1B S 58 Write Transaction 20 1691870 1020.874.649 657 us 1B S 58 Read Transaction 11+ 1691871 1020.874.629 81.6 us 1B S 58 Write Transaction 23 1691872 1020.874.710 29.1 us 0 B SP 58* Read Transaction 23 * * * * * * * * * * *<	1691868	10:20.874.384	56.6 us	1 B	SP	58	Read Transaction	4F*			
1691870 10220.874.549 56.7 vs 18 SP 58 € Read Transaction 11* 1691871 10220.874.529 816.0 vs 18 S 58 VM ransaction 23 1691872 1020.874.710 29.1 vs 08 SP 58 € Read Transaction 23 ₹ Text ↓ We Search ↓ W W	1691869	10:20.874.467	82.4 us	1 B	S	58	Write Transaction	20			
1691871 1020.874.829 81.6 us 1 B S 58 Write Transaction 23 1691872 10.20.874.710 29.1 us 0 B SP 58 ● Read Transaction ✓ Text Q LiveSearch ✓ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1691870	10:20.874.549	56.7 us	1 B	SP	58	Read Transaction	11*			
1691872 10.20.874.710 29.1 us 0 B SP 58' Read Transaction	1691871	10:20.874.629	81.6 us	1 B	S	58	Vite Transaction	23			
C > Text 4 UbeSearch 0 MoRRAR 1000 Macanda Endered lace	1691872	10:20.874.710	29.1 us	0 B	SP	58*	Read Transaction		~		
	<								>		
No Silver 1 602 M records	Text • 9, LiveS	earch 🗸 🖬 📔							+ - 🖬 🖬 🗄 🗮 🕽		
No muei: 1.092 Princeolos. IZC Into Bus Liveriter Into	No filter: 1.692 M record	ds.							Protocol Lens: I2C 🔹 🖮 👻	Bus LiveFilter Info	

Total phase Aardvark I2C Protocol analyzer.

A simple tool that tracks every transaction.

Expects i2c clock and data lines connected.

Snoops on that two lines. Helps catch the buggy transaction. Ex: The one in the end has no data.



Debug Summary

- We Don't need a sledge hammer to break open the eggs.
- Start with simple prints, sysfs entries & software debug tools.
- Only if the kernel is crashing get on to multimeter & oscilloscope.
- I2c protocol analyzers only when we suspect a hardware bug.



Thank You



References

- TPS65218 Public Data Sheet: <u>http://www.ti.com/lit/ds/symlink/tps65218.pdf</u>
- http://www.ti.com/lit/ug/sprui50/sprui50.pdf
- https://www.kernel.org/ Linux Mainline kernel
- <u>http://events17.linuxfoundation.org/sites/events/files/slides/schulz-pmics-keep-power-in-your-hands.pdf</u>
- <u>https://www.motioncontrolonline.org/blog-article.cfm/LDOs-vs-SMPS-Power-Supplies-for-Industrial-Automation-Equipment/44</u>
- <u>https://manpages.debian.org/unstable/i2c-tools/i2cset.8.en.html</u>



Back Up slides

- -f: Force access to the device even if it is already busy. By default, i2cset will
 refuse to access a device which is already under the control of a kernel driver.
 Using this flag is dangerous, it can seriously confuse the kernel driver in
 question. It can also cause i2cset to silently write to the wrong register. So use
 at your own risk and only if you know what you're doing.
- -y: Disable interactive mode. By default, i2cset will wait for a confirmation from the user before messing with the I2C bus. When this flag is used, it will perform the operation directly. This is mainly meant to be used in scripts.

