Understanding, Utilizing and Choosing CubeSat Kit™ PPMs

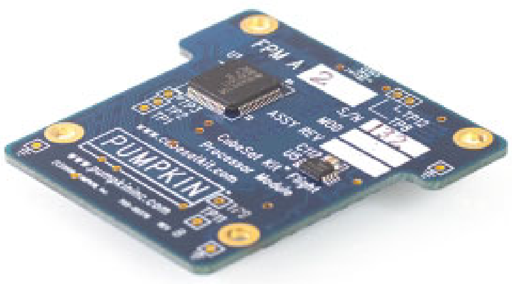
Andrew E. Kalman, Ph.D.

Outline

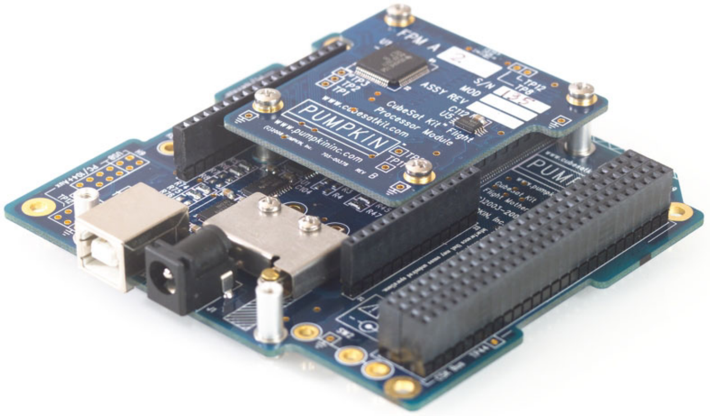
- Part I: What is a PPM?
- Part II: Why is a particular PPM designed this way?
- Part III: Which PPM works for my CubeSat?

I: PPMs Explained

- A Pluggable Processor Module (PPM) is a small hardware module that maps a particular microcontroller (MCU) – with its unique I/O, peripherals and features – into the CubeSat Kit (CSK) architecture:
  - on the Motherboard (MB), where the radio, USB, RTC, SD Card etc. are, and …
  - on the CSK bus, the CubeSat Kit's "backbone"

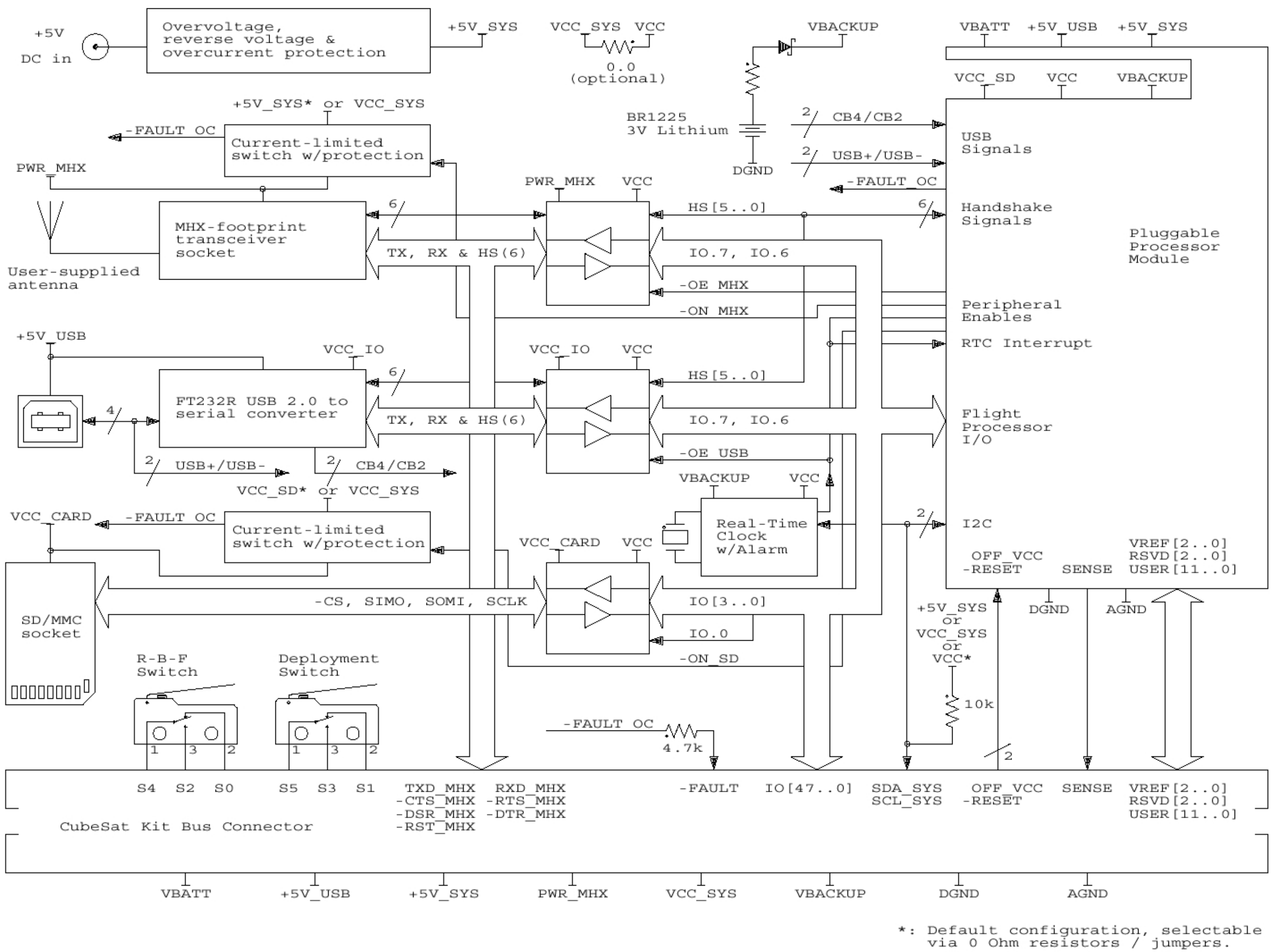


→



PPM w/100-pin PPM connector | mates with … | MB w/100-pin PPM connector

CSK Architecture Block Diagram



PPM Connector

H10
LSS-150-02-L-DV

<->	IO.23	1	2	IO.47	<->
<->	IO.22	3	4	IO.46	<->
<->	IO.21	5	6	IO.45	<->
<->	IO.20	7	8	IO.44	<->
<->	IO.19	9	10	IO.43	<->
<->	IO.18	11	12	IO.42	<->
<->	IO.17	13	14	IO.41	<->
<->	IO.16	15	16	IO.40	<->
<->	IO.15	17	18	IO.39	<->
<->	IO.14	19	20	IO.38	<->
<->	IO.13	21	22	IO.37	<->
<->	IO.12	23	24	IO.36	<->
<->	IO.11	25	26	IO.35	<->
<->	IO.10	27	28	IO.34	<->
<->	IO.9	29	30	IO.33	<->
<->	IO.8	31	32	IO.32	<->
-->	IO.7 *	33	34	IO.31	<->
<--	IO.6 *	35	36	IO.30	<->
-->	IO.5	37	38	IO.29	<->
<--	IO.4	39	40	IO.28	<->
<--	IO.3 *	41	42	IO.27	<->
-->	IO.2 *	43	44	IO.26	<->
<--	IO.1 *	45	46	IO.25	<->
<--	IO.0 *	47	48	IO.24	<->
	+5V USB	49	50	+5V USB	
	+5V_SYS	51	52	+5V_SYS	
	VCC SD	53	54	VCC SD	
	VCC	55	56	VCC	
	DGND	57	58	DGND	
	AGND	59	60	AGND	
	VBATT	61	62	VBATT	
	VBACKUP	63	64	VBACKUP	
	VREF0	65	66	-FAULT OC	-->
	VREF1	67	68	SENSE	-->
	VREF2	69	70	-RESET	<--
	RSVD0	71	72	OFF VCC	<--
	RSVD1	73	74	SDA SYS	<->
	RSVD2	75	76	SCL SYS	-->
-->	USBDP/CB4	77	78	USER0	
-->	USBDM/CB2	79	80	USER1	
<--	-ON SD	81	82	USER2	
<--	-ON MHX	83	84	USER3	
<--	-OE MHX	85	86	USER4	
<->	-OE USB/-INT	87	88	USER5	
-->	HS0	89	90	USER6	
-->	HS1	91	92	USER7	
-->	HS2	93	94	USER8	
<--	HS3	95	96	USER9	
<--	HS4	97	98	USER10	
<--	HS5	99	100	USER11	

The CubeSat Kit PPM connector on the PPM and the MB has I/O, power, control, status, I2C, handshaking and user signals.

The forty-eight I/O signals on the PPM are the same as those on the CSK Bus.

Eight power signals route power between the PPM and the MB.

The remaining forty signals are for control, status, I2C, handshaking, user-defined, etc.

The entire CubeSat Kit Bus connector (except for S[5..0] & MHX socket signals) is available to PPM.

PPM-to-CSK Mapping

- Dedicated / Mandatory:
  - IO.[3..0]: the first SPI interface (SPI0)
  - IO.[5..4]: the first UART interface (U0)
  - IO.[7..6]: the second UART interface (U1)
  - SCL_SYS & SDA_SYS: the first I2C interface (I2C0)
  - Control signals (e.g., –ON_SD)
- Recommended / Optional:
  - IO.[47..40]: for analog voltages AN.[7..0]
  - IO.[39..32]: for analog voltages AN.[15..8]
  - VREF0|1|2: for analog reference voltages
  - Handshake signals (e.g., HS[5..0])

II: Understanding a particular PPM

- Ideally, map any peripheral function (e.g., I/O, serial, analog) to any PPM signal.
- Generally speaking, only ASICs ($$$$) and FPGAs ($$) can do this. MCUs are much cheaper, and simpler to use.
- When mapping an MCU to a PPM, strive for a sensible allocation of MCU resources to the PPM signals, according to:
  - Satisfy MCU-compatible power / POR / BOR / reset / programmer / debugger requirements
  - Assign available peripherals to mandatory features (e.g., UARTs to U0 & U1)
  - Assign available peripherals & I/O to optional features
  - Assign remaining I/O intelligently to CSK Bus I/O
  - Implement additional PPM features with leftover MCU resources

PPM implementation by MCU

PPM	Signals	Comment
A1 MSP430F161x 64 pins (48 I/0)	I/O	All ports P1-P6 mapped to IO.[47..0]
	Analog	VREF[2..0] utilized; AN.[7..0] on IO.[47..40] (shared w/handshaking)
	Control	Shared with P1.7, P4.6, P4.5 & P6.6
	Handshake	Shared with P6.0, P6.1, P6.2, P6.3, P6.4, P6.5
	Feature	I2C isolator to handle I2C0-SPI0 conflicts
B1 C8051F120 100 pins (64 I/0)	I/O	P0-P3 mapped to IO.[29..0]; IO.[33..31] not implemented
	Analog	VREF[2..0] utilized, AIN0.[7..0], CP[1..0] & DAC[0..1] on IO.[47..34]; no VREF reference output available
	Control	Via dedicated P4
	Handshake	Via dedicated P5
	Feature	128KB SRAM (via P4, P6 & P7)
D1 PIC24FJ256GAx 100 pins (85 I/0)	I/O	48 IO ports mapped to IO.[47..0]
	Analog	VREF[2,0] utilized, AN.[15..8, 7..0] on IO.[39..32, 47..40], shared w/I/O
	Control	Via dedicated I/O
	Handshake	Via dedicated I/O
	Feature	64Mbit serial Flash memory, U2, U3, SPI1, SPI2, I2C1 & I2C2 free and mapped to CSK Bus

PPM A/B/D Commonality

- Get power from +5V_SYS and/or +5V_USB
- Provide VCC & VCC_SD @ +3.3V
- External Reset / POR / BOR circuit
- External OFF_VCC input
- External SENSE and -FAULT_OC outputs
- USER[11..0] untouched

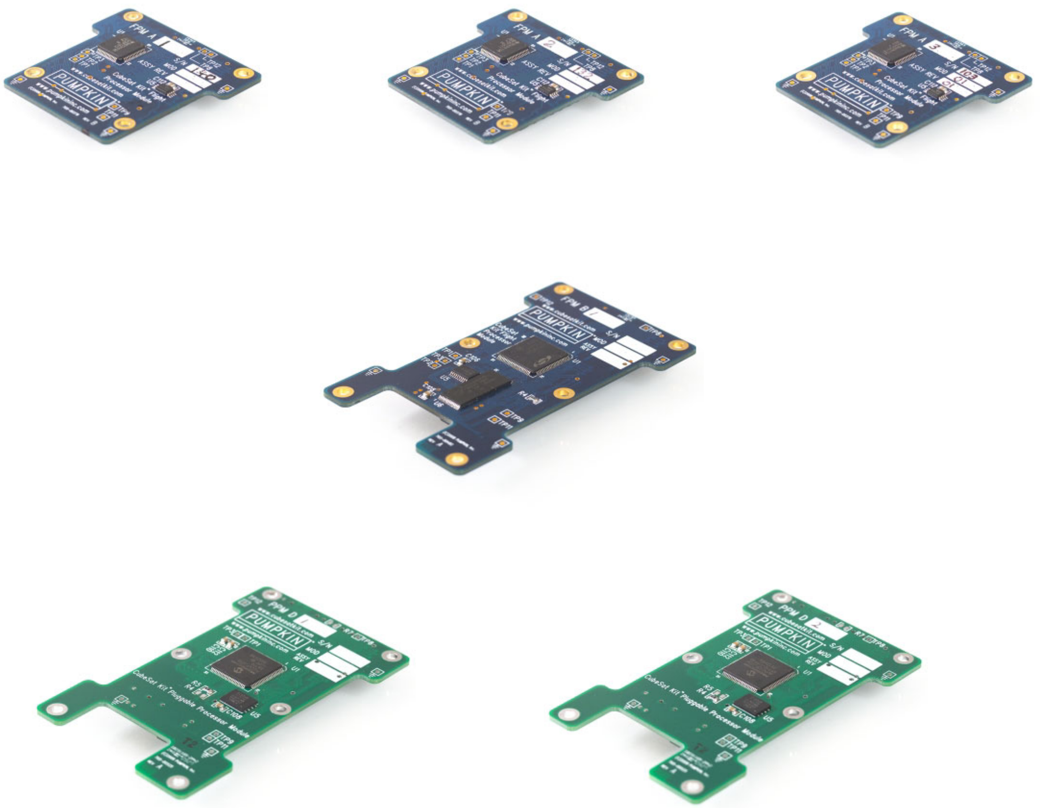
III: The Right PPM for the Job

- Is the MCU compatible with your code?
- Does the MCU have the features / speed / power you want / need?
- Will shared signals work in your design?
- Do you like the compatible tools (e.g., IDEs, compilers, debuggers)?

- MCU migration is usually within families, so invest in a MCU architecture wisely.

Conclusion

- Purely in terms of I/O and CSK standard peripherals (i.e., SPI0, U0, U1 & I2C0), PPMs are nearly interchangeable from a hardware perspective.
- MCU-specific differences are reflected in functions of signals IO.[47..8], and MCU internal capabilities.
- Dedicated control & handshaking is better than sharing I/O pins, but requires more than 48 I/O pins.
- Surplus pins are used on-board PPM to implement additional general-purpose features attractive to all CSK customers.
- Mission-specific features should be implemented through the MCU's I/O space via the CSK Bus.



Q&A Session

Thank you for attending this Pumpkin presentation at CubeSat Developers Workshop 2009!

Notice

This presentation is available online in Microsoft® PowerPoint® and Adobe® Acrobat® formats at:

www.pumpkininc.com/content/doc/press/Pumpkin_CSDWLU_2009.ppt

and:

www.pumpkininc.com/content/doc/press/Pumpkin_CSDWLU_2009.pdf

Appendix

- Speaker information

  - Dr. Kalman is Pumpkin's president and chief technology architect. He entered the embedded programming world in the mid-1980's. After co-founding Euphonix, Inc – the pioneering Silicon Valley high-tech pro-audio company – he founded Pumpkin, Inc. to explore the feasibility of applying high-level programming paradigms to severely memory-constrained embedded architectures. He is the creator of the Salvo RTOS and the CubeSat Kit. He holds two United States patents. He is a consulting professor in the Department of Aeronautics & Astronautics at Stanford University and directs the department's Space Systems Development Laboratory (SSDL). Contact Dr. Kalman at aek@pumpkininc.com.

- Acknowledgements

  - Pumpkin's Salvo and CubeSat Kit customers, whose real-world experience with our products helps us improve and innovate.

- CubeSat Kit information
  - More information on Pumpkin's CubeSat Kit can be found at http://www.cubesatkit.com/.
- Copyright notice

First presented at the CubeSat Developers Workshop in Logan, Utah at Utah State University on Sunday, August 9, 2009.