Thermal Instrumentation Card (TIC) Module Firmware

1.1 Abbreviations:

CCMM :   Chassis Control Microcontroller Module

ADC: ADS124S08 Analog-to-Digital Converter

MSP430:   MSP430G2553 Microcontroller

ISR: Interrupt Service Routine

LPM0:   Low-Power Mode

1.2 Hardware Pinout:

MSP430G2553			
Name	**Pin**	**I/O**	**Description**
DIn	P1.1	I	SPI data in
Dout	P1.2	O	SPI data out
SCLK	P1.4	O	SPI clock
!CS	P1.5	O	Chip select [active low]
SCL	P1.6	I	I2C Clock
SDA	P1.7	I/O	I2C Data
I2CA0	P2.0	I	I2C address bit 0
I2CA1	P2.1	I	I2C address bit 1
I2CA2	P2.2	I	I2C address bit 2
!RST	P2.3	O	Reset ADC
START/SYNC	P2.4	O	Start data conversion (not used by firmware)
!DRDY	P2.5	I	Conversion data is ready (not used by firmware)

Table 1

1.3 Summary:

The TIC module firmware is an interface between the CCMM and ADS124S08
ADC. The CCMM sends single-byte commands to an MSP430G2553
microcontroller via I2C bus, which can contain instructions to run
procedures for ADC reset, self-offset calibration and burnout detect.
In response, the MSP430 runs each of the prompted procedures, if any,
and stores the ADC data and status flags in a 40-byte output buffer.
The CCMM can then read this 40-byte buffer over the I2C bus. The
MSP430 communicates with the ADC via an SPI bus and three dedicated
pins (!RST, START/SYNC and !DRDY). It is an SPI master, and an I2C
slave device.

 Upon power-on, the MSP430 performs the following startup tasks:

- Stop watchdog timer.
- Initialize clock to 8 MHz
- Initialize pin I/O direction and polarity as specified in Table
  1, select SPI and I2C pins.

- Initialize SPI for rising-edge clock, master mode, MSB first, 8-bits data and synchronous mode. Configure baud rate to 9600 and enable SPI interrupts.
- Initialize I2C for synchronous mode, set slave address by reading I2C address pins, and enable I2C interrupts.
- Run ADC reset, self-offset calibration, and burnout detect procedures (see details in section 1.5)
- Read ADC registers 0x00 through 0x09 and store in array.
- Enter low-power mode.

Upon entering low-power mode, the MSP430 waits to detect an I2C receive interrupt from the CCMM. When an I2C receive interrupt is detected, the MSP430 reads a single byte transmitted by the CCMM, and then parses it for command bits as detailed in table 2:

CCMM Command Byte	
Bit	**Instruction**
Bit 0	ADC Reset
Bit 1	Run Self Offset Calibration
Bit 2	Run Burnout Detect
Bits [3:7]	No Command

Table 2

For each instruction received in the command byte, if any, the corresponding procedures are executed in the following order: ADC Reset, Self-Offset Calibration, then Burnout detect. After all commanded procedures have been executed, if any, the MSP430 changes the channel read by the ADC and requests 3 bytes of data for the following inputs: channels 0-5, analog source voltage, digital source voltage and on-chip temperature. Each set of data is stored in a corresponding 3-byte register. See section 1.5 for details on the ADC Reset, Self-Offset Calibration, Burnout Detect, and ADC Read procedures.

Once data from all channels have been collected, the MSP430 copies the data and status bytes into a 40-byte output buffer, which is sent to the CCMM when it reads from the MSP430. The output buffer format is described in table 3.

MSP430 Output Data Buffer	
Bytes	**Details**
[0:3]	[0:2] Channel 0 Data [3] spare status byte
[4:7]	[4:6] Channel 1 Data [7] spare status byte
[8:11]	[8:10] Channel 2 Data [11] spare status byte
[12:15]	[12:14] Channel 3 Data [15] spare status byte
[16:19]	[16:18] Channel 4 Data [19] spare status byte

[20:23]	[20:22] Channel 5 Data [23] spare status byte
[24:27]	[24:26] AVDD Data [27] spare status byte
[28:31]	[28:30] DVDD Data [31] spare status byte
[32:35]	[32:34] Temperature Data [35] spare status byte
[36:39]	[36] [0] Ch0 burnout [1] Ch1 burnout [2] Ch2 burnout [3] Ch3 burnout [4] Ch4 burnout [5] ADC Reset performed [6] ADC self-offset calibration performed [7] ADC burnout detect performed [37:39] spare status bytes

Table 3

Once the entire output buffer has been sent, the MSP430 exits low-power mode, re-initializes the I2C configuration in case the master is stuck, re-enters low-power mode and waits for another I2C Rx Interrupt.

1.5 Functional Description:

I2C State Machine:

The MSP430 handles communication with the CCMM using an interrupt- driven I2C state machine. The MSP430 is an I2C slave in synchronous mode, with a native address of 0x00. The address is slave address is set when the MSP430 is powered on by reading I2C address pins (see table 1).

Start, Stop, and NACK interrupts are detected by the USCIAB0RX vector, shared with the SPI ISR. Received and transmitted I2C data interrupts are detected by the USCIAB0TX vector.

The initial state is I2C_RX_REG_ADDRESS_MODE.

- In this state, the MSP430 waits in low-power mode to receive a byte from the CCMM. When an I2C RX interrupt occurs, the MSP430 reads the first received byte and parses it for command bits with instructions for the ADC Reset, Self Offset Calibration, and Burnout Detect procedures. The format of the command byte is shown in table 2.
- For each instruction received in the command byte, if any, the corresponding procedures are executed in the following order: ADC Reset, Self-Offset Calibration, then Burnout detect. The state is then set to I2C_TX_DATA_MODE, and the Read ADC Data procedure is performed. The data and status information are then published to the data output buffer, and the MSP430 exits low-power mode.

In I2C_TX_DATA_MODE:

- When an I2C TX interrupt occurs, the MSP430 adds the first byte in the data output buffer to the SPI TX buffer, increments the index of a counter and waits for another I2C TX interrupt, adding the next byte to the data output buffer. This repeats until all 40 bytes have been requested by the CCMM. The MSP430 then returns to I2C_RX_REG_ADDRESS_MODE and waits in LPM0 for an I2C RX interrupt.

SPI State Machine:

The MSP430 handles communication with the ADC through an interrupt-driven I2C state machine. The MSP430 is an SPI master, configured for rising-edge clock, MSB first, 8-bits data, synchronous mode and 9600 baud.

SPI commands sent by the MSP430 to the ADC are defined by functions that transmit one or more bytes and may receive zero or more bytes. Two counters are used to keep track of the number of bytes that need to be transmitted and then received, respectively. Each command function starts communication by pulling the CS pin low and sending its first TX byte to the ADC, before entering LPM0

The initial state is IDLE_MODE.

- This is the default state. When the MSP430 is sending commands to the ADC, it will transition to TX_DATA_MODE.

TX_DATA_MODE:

- If there are any bytes left to transmit, the MSP430 adds its next TX byte to the SPI TX register.
- If there are no bytes left to transmit, but one or more bytes to receive, the MSP430 switches to RX_DATA_MODE, delays 25 us, and then adds the current value of the SPI RX register to the RX buffer and increments the RX counter.
- If there are no bytes to transmit or receive, the state returns to IDLE_MODE and the MSP430 exits LPM0.

RX_DATA_MODE:

- If there are bytes left to receive, the MSP430 adds the current value of the SPI RX register to the RX buffer and increment the RX counter. The MSP430 also sends a dummy byte of 0x00 to keep the clock cycling.
- If the expected number of bytes have been received, the state returns to IDLE_MODE and the MSP430 exits LPM0.

Received data is then copied into a dedicated buffer (eg Channel 1 ADC data).

Nominal Delays:

(these delays assume a clock calibrated to 8 MHz)

Delay	Description	Time
t_{conv}	Start Conversion	125 ms
t_{sfocal}	Self-Offset Calibration	600 ms
t_{rst}	Reset	125 ms

Table 4

Procedure	Delay
Power-On Startup	1725 ms
ADC Reset	$2 * t_{rst} + t_{conv} = 375$ ms
Self Offset Calibration	$t_{sfocal} = 600$ ms
Burnout Detect	$6* t_{conv} = 750$ ms
Read ADC Data	$10 * t_{conv} = 1250$ ms

Table 5

ADC Reset:

The ADC Reset pin is pulled low, and then the MSP430 delays for t_{rst} before pulling the Reset pin high again. The MSP430 delays for a further t_{rst}, and then sends start conversion command (0x09) and writes to the ADC's DATARATE register to enable global chop, before delaying for t_{conv}.

Self Offset Calibration:

The MSP430 sends the Self Offset Calibration command (0x19) to the ADC and delays for t_{sfocal}.

Burnout Detect:

The MSP430 makes three register write commands to the ADC in order to enable PGA, set gain to 32 V/V, turn off global chop, and turn on burnout current sources.

The MSP430 then writes to the INPMUX register to set it to measure Channel 0 (voltage from Ain0 to Ain1) , followed by a delay of t_{conv}. The MSP430 then sends a read data command to the ADC and receives 3 bytes. If they are EQUAL to the expected full-scale value 0xFFFF7F, a burnout is assumed and the status bit is set (see table 3). This

process is repeated for Channels 1, 2, 3 and 4. The MSP430 then instructs the ADC to turn off burnout current sources. A delay of t_{conv} follows.

Read ADC Data:

The MSP430 makes three register write commands to the ADC in order to enable PGA, set gain to 32 V/V, turn off global chop, and turn off burnout current sources.

The MSP430 then writes to the INPMUX register to set it to measure Channel 0 (voltage from Ain0 to Ain1) , followed by a delay of t_{conv}. The MSP430 then sends a read data command to the ADC and receives 3 bytes, and these bytes are stored in a corresponding data-holding buffer. This process repeats for Channel 1, 2, 3, 4, 5, the analog voltage monitor, digital voltage monitor and on-chip temperature sensor. For the voltage monitors and on-chip temperature sensor, the MSP430 writes to the ADC SYS register instead of INPMUX. Furthermore, the ADC PGA gain must be set to 4 before reading the on-chip temperature sensor.

After all read commands have been stored in their respective data-holding buffers the MSP430 instructs the ADC to turn off burnout current sources (they should not be on at this point) and delay for t_{conv}.