**Rev 11 Aug 2017**

**Using the Byonics MTT4B for SATT4 low-risk Testing/Education**

This paper describes how to use a stock MTT4B from Byonics.com to simulate the SATT4 for student familiarization and experiments. It details the few differences between the stock MTT4B and the flight unit, SATT4.

The design goal of the SATT4 board is to be a low cost AX.25 and APRS compatible Cubesat comms board for telemetry, command, and control and to encourage Amateur Radio missions on student cubesats compatible with the worldwide network of volunteer ground stations.

Although the board parts cost about the same as the stock MTT4B from Byonics, after the student labor to build, test, and troubleshoot it represents a ***huge*** investment and the value goes up an order of magnitude.

Figure 1. The SATT4 flight board:

It is better to bring students up to speed with this flight ready board by having them build all prototype and familiarization experiments using an off-the-shelf Byonics MTT4B rather than risk the flight board at an early stage.

***Refer to the SATT4 documentation for the full description of the Flight Board system.***

**SATT4/MTT4-B Specifications:** The MTT4B is identical to the SATT4 in all of the noted SATT4 capabilities except as noted below in red.

* An APRS Digipeater function to relay user packets from uplink to downlink
* Design Operation from a 7 volt bus up to 12 volts max.
* Five analog inputs for simple telemetry in a 1-minute beacon
* 8 digital output bits that can be toggled by ground control for spacecraft functions.
* A serial port for uplink and downlink between other boards and the ground
* X - A CANSAT 9600 baud high speed UHF downlink for data
* X - A Heartbeat watchdog timer that power cycles the board if Telemetry every fails



Figure 2: Comparison of stock MTT4B and SATT4 block diagram



Figure 3. MTT4B serial ports, analog telemetry inputs and command outputs

**Preparation of MTT4B for SATT4 Simulation and Testing:** Do these preparations to the stock MTT4B to make it ready for programming identical to the SATT4:

1. First, prepare connectors for the serial ports and for the Command bits as shown in Figure 3.

1. Solder a female pin header for the analog header as shown here.

**Figure 4.** The top analog female header

1. Next add the 8.25k pullup and 27.4k pulldown resistor for the $20 bank select command bit on the bottom of the board as shown here and in the schematic in appendix A.

You can add the actual thermistor off the board later via the analog Ch5 pins when you do the telemetry experiments. You can also install these resistors off the board also, but if you do, then the MTT4B will not come up in the correct BANK1 default unless the external resistors and thermistor are plugged in.

 **Figure 5.** The bottom $20 command bit mod:

These thermistors do not need to be precision resistors and a common 8.2k and 27k resistor can be used since you will probably want to do your own calibration run for whatever 30k thermister you end up using.

**Dual Bank Switching: The** SATT4 and MTT4B both **have** dual configuration banks to program setting for two separate personalities. For APRS satellites, we use this switchable configuration to simply turn on and off the user Digipeating mode. We program the default bank(1) as the (safe mode) with the digipeater off and program the commanded bank (0) to be with the digipeater on.

Since this dual use of the Ch5 telemetry for both temperature and BANK control, then you will need to do TWO calibration runs on your thermistor. One with the $20 bit on and the other with it off. By doing both curves then you can always get temperature data independent of bit $20,

**Bank 0 - TOCALL=APDIGI: normal, Digipeater ON, TLM Ch5 is biased high**

**Bank 1 - TOCALL=APOFF: Bootup mode with minimum power. Ch5 is valid.**

**Warning:** Until you install the pull-down thermistor on the AN5 and $20 bit resistor, the power up default of the MTT4B is bank0. But once these components are installed the default operating bank becomes bank 1. But this does not apply during programming! Be aware of this default to bank 0 always while programming and don’t screw up the settings in the wrong bank.

REWRITE THE SAATT$ manual for POITIVE Logic!

**SATT4 Telemetry:** The MTT4B analog telemetry inputs are available on a single row of solder pins as shown above. Students should build a set of telemetry sensors as shown in the schematic here. The calibration for the Tbatt sensor will be a different curve because it has the special bias to work with the $20 command bit. The current sensor we use on our spacecraft is no longer available, so students should explore another current sensor for this flight.

**5V Regulators:** The SATT4 contains a linear 5v regulator since it was designed to operate form a 6 to 7 volt bus. The MTT4B has a 5V switching regulator designed to operate as high as 14 volts input.

**High Speed UHF Downlink:** The MTT4B does not have the SATT4 board’s optional UHF 9600 baud downlink CANSAT transmitter made by Stensat with an added UHF power amplifier. Notice also that the 9600 baud UHF downlink has nothing to do with APRS nor the global network of APRS ground stations that only listen on VHF.

**SATT4B Uplink Command BITS:** The SATT4B has a command BYTE consisting of 8 uplink command configurable output bits ($XX). The $02 bit cannot be used as it is used internally. The $20 bit is also used for the bank-select bit but it can be also used to signal that state to some other devices. The first thing students should build is an LED status indicator for testing the command link. Connect 7 LED’s as shown here to indicate the status of the command bits.

**Configuring the SATT4 Settings:**

**Serial Port**: A big difference between the SATT4 and the stock MTT4B is that the MTT4B has true RS232 serial interfaces and can plug directly into a PC serial port for programming and for operation. The SATT4 serial interfaces require an external TTL to RS232 converter as shown here.

**Dual Serial Ports:** On the MTT4B the A port is used for firmware loading and general serial communications and the B port was designed in the original MTT4B for connection of a 4800 baud GPS. We do not use GPS on a cubesat, so we configure this second port to operate at 9600 baud as a second serial text data line using the BMODE = TEXT and BBAUD = 9600 commands.

On the MTT4B, the port A serial port for serial communications and configuring is on the sub miniature phone jack. Use the programming cable that came with the MTT4B to access that port. The DB9 serial port is actually port B and is configured to require a NULL modem cable to connect to the PC. It turns out that the MTT4B can also be configured on this port as well.

Appendix A: Unmodified schematic of MTT4B and the modified schematic for the SATT4B





**\* - FOUO - \***

**Appendix B**. Setup and Programming commands for SATT4 Processor v0.73. The SATT4 has two banks for two different flight configurations. We normally configure the BANK0 for full flight operating mode with the User Digipeater enabled and BANK1 for minimum power mode with just telemetry and no user digipeating. The banks can be commanded via the $20 command in flight.

Command Default BANK0 BANK1 (same as Bank0 except…)

---------------- --------------- ------------------------------- ----------------------------------------------

MYCALL NOCALL TJHS TJHS-1

ABAUD 19200 9600 \* (\* means same)

ALIAS1 TEMP APRSAT none (or secret)

ALIAS2 % ARISS none

ALIAS3 % WIDE none

ALTNET APTT4 APDIGI APSAFEI

AMODE Text text \*

BANK 0/1 0 0 1 🡸 set during programming

BBAUD 4800 9600 \*

BMODE GPS text \*

BNKMODE 0 1 \* 🡸 uses JP1 for bank switching

 [uses PA3, CPU pin 37, J3?]

BPERIOD 0 175 \* <a 3 minute beacon text>

BTEXT >TT4 alpha >BANK0 Beacon text >BANK1 Beacon Text

CDMODE Tones \*\*\*Inverted -pin2 \* <J1 pin 2 [CPU pin 20] to holdoff XMT>

DIGIID true true \* <inserts MYCall in digipeated packets>

DIGIMY false TRUE \*

DUPETIME 30 10 \*

HEADERLN false false \*

LEDS true true \*

MSGCMD false true \* <allows !OUT $xx commands to BS2 CPU >

NODISP false true \* <Required for $xx PortC msg commanding>

PATH1 WIDE1-1 ARISS (via ISS) \*

PATH2,3 WIDE2-1 (none) \*

PKTICOM true true \* <Enables received packets to PC text port>

PKTOCOM False true \* <Lets serial port see our TX packets also>

PPERIOD 0 0 \* <posit Prd (In sec), not used in satellites

RXAMP 5 5 5 \* <RX gain. Use MONITOR mode to adjust>

TELHIRES true true \* <enables 0-999 values>

TELVOLT true False \*

TELTEMP true False \*

TELREAD true False \* <Turns off human readable telemetry>

TPERIOD 0 60 \* <Telemetry Period >

TXD 40 30 \* <20 = about 150ms>

TXLEVEL 128 128? \* <TBD for flight>

TXTWIST 50 50? \* <TBD for flight>

TXFREQ 144.39 145.825 \* <!>

RXFREQ 144.39 145.825 \* <!>

RXSQL 0 1 2 <tighter squelch>

RXSAVE 0 1 1 <turn on for power saver>