Rocketometer Operator’s guide

Chris Jeppesen, 24 Sep 2013

1) Mechanical

The user interface includes two buttons and one power switch. The buttons include one reset button, and one button intended to switch from USB Mass Storage readout to main Rocketometer mode (boot). The reset button has been disabled by removing the solder bridge next to it. The boot button doesn’t actually do anything, so it doesn’t need to be rendered inert.

The power switch should be glued or otherwise secured in the ON position, with the switch closer to the word ON and closer to the edge of the board.

The SD card should also be glued into place, with something that can be peeled off to remove the card after flight.

Some components on the board are very close to the mounting holes. They are technically out of the keepout zone for the screw head, but too close for comfort. It would be great if rubber, paper, or some other non-conducting washers could be used to protect these components.

2) Indicator lights

The board has four colors of lights, yellow, red, green, blue.

The yellow light is on whenever the charging circuit is charging the battery. It is off when there is no external power OR when the battery is fully charged.

The RGB diode is green for the first few seconds during board startup. It then flickers through all of the colors slowly when the board is not vertical and has not been for 20 minutes. It flickers quickly when the board has been vertical within the last 20 minutes. To be precise, each time it writes a sector to the SD card, it toggles one of the colors on or off, and it cycles through the colors in turn. Since more data is generated in vertical mode, sectors are written more quickly, so the light blinks more quickly.

When the board is in slow recording mode, it flickers one light on, all off, next light on, etc, with the light on for noticeably less than 1 second. When the board is in blinklock (see Trouble below), the light is on for exactly 1 second, then off for 1 second, etc. This is noticeably slower than slow recording mode. It takes a bit of care to notice the difference, but no special tools other than the Mark I Eyeball.

3) Power

The board has a +5V input, which it uses to run all functions and charge the battery while powered. When unpowered, the board automatically and seamlessly switches to battery power. This board, with slightly different software, has been observed to run for about 2.5 hours before draining the battery. This means that the board has a power draw of significantly less than 100mA. The battery is rated at 110mAh.

The +5V input also charges the battery. The board carries the proper LiPo charging circuit and the battery has an integrated LiPo protection circuit which among other things prevents overcharge and overdischarge. Charging current is limited to 100mA, which has been observed to fully charge the battery in about 1.5 hours.

4) Trouble

If the board detects an unrecoverable error, it will enter a mode called blinklock. In this mode, the RGB diode blinks slowly, 1 second on, 1 second off. The blink is a simple binary code. The error number is communicated by blinking green for a 1 bit and red for a 0 bit, least significant bit first. When all 1 bits have been blinked, the blue light blinks indicating that the full number has been blinked out, then the process repeats, blinking the same number forever until the device is reset or power-cycled.

Let’s say the blink code is 404. In binary this is 1\_1001\_0100, so the code would be R R G R G R R G G B endlessly repeated. The first blink after a blue light is the first blink in this cycle.

There is no table of blink codes except for the source code. The code would have to be examined to determine the exact cause. If you encounter a blink code, notify me and disable power (and battery if possible, otherwise you have to wait for the battery to drain) to reset the board.

5) File generation

The Rocketometer writes files in the standard FAT32 filesystem, with some tweaks. In particular, the card has been formatted with the largest possible cluster size and only one file allocation table, to optimize writing speed. Any normal operating system should be able to read this card with no issue, and it has been tested with standard Microsoft Windows 7 and Linux, which did read it with no issue.

Files are named in the format RKTO%04d.SDS, where %04d indicates four decimal digits. These digits are the serial number for the file.

Each time the board starts up, it will search the card, starting with a serial number of 0 and incrementing it by 10 each time. When it does not find a file corresponding to this serial number, it stops searching and creates a new file with this number. In this case, if the card is repeatedly restarted, it will count

RKTO0000.SDS

RKTO0010.SDS

RKTO0020.SDS

etc.

Files are capped at 1GiB. When the current file size reaches this amount, the file is closed. The card is searched again starting at the next higher serial number and incrementing by 1 each time.

At high data rate, the card will hit this 1GiB file size in about 1 day. If the card ran for 10 days straight starting with a serial number of 0, it would reach serial number 10. This still isn’t a problem, when the card is reset it will skip that 10 and go on to the next multiple. No data will be overwritten.

The root directory can hold 2048 files. It is unlikely that this number will be reached, as it would involve being power-cycled that many times.

Since this is a 16GiB card, it can record for about 16 days at high rate and 160 days at low rate. There should be no problem with either number of files or data volume.