

Klondike Technical Reference

This document describes technical details and operation of the Klondike Bitcoin miners.

There are currently three related but different designs:

- K1 (Nano) - a small USB dongle containing one Avalon ASIC.
- K16 - a medium size 10cm x 10cm board containing 16 Avalon ASICs.
- K64 - a 4-up chained panelization of the K16 design, 20cm x 20cm.

These designs are based on a PIC16LF1459 micro-controller with USB connectivity. This device is low-cost and has enough functionality to handle all the tasks required.

The following PIC hardware functions are utilized in the Klondike:

- USB engine – with software stack support, provides for host control.
- I2C engine – simple bus to communicate with chained boards.
- USART – used to receive result data from the ASICs on board.
- Timer0 – provide timing for ASIC work progress and housekeeping.
- Timer1 – measure fan tachometer output for fan speed (K16,K64).
- Timer2,PWM – provide fan speed control (K16,K64).
- Temperature Sensor – for monitoring successful heat dissipation.
- ADC – for measuring an external thermistor, as above (K16, K64).
- I/O pins – send data to the ASIC chain, enable clock, drive an LED indicator.

Upon device start up the PIC is programmed to attach to a USB host as a device with serial emulation. If it cannot detect a USB host then it falls back on I2C for host control. It monitors both these channels until it can begin operation which consists of accepting commands from it's host master. These commands are described in detail in this document.

In addition to the PIC controller the Klondike board also contains suitable regulated power and primary clock generation for the ASICs on board. This consists of 1.2V core power, 3.3V I/O and PIC power, and 1.2V clock generator power. The 1.2V core and 3.3V power are provided using buck regulator switching technology. The 1.2V clock generator supply is provided by a linear regulator to isolate it from the core supply. The clock is provided by a high precision oscillator package, or optionally from the internal PIC clock generator.

The K1 design uses a buck regulator only for the 1.2V core supply. The other two supplies are provided by linear regulation. All three are derived from the USB 5V supply. The K16 / K64 source power from a 12V ATX type power supply.

PIC16LF1459 Pin Allocation

The following table indicates the use allocated to each PIC I/O pin.

| Pin# | Name | Klondike Usage |
|------|------|---|
| 19 | RA0 | USB D+ |
| 18 | RA1 | USB D- |
| 4 | RA3 | Fan tachometer sense input (also MCLR to program) |
| 3 | RA4 | CLK_EN enable ASIC clock (optionally CLK_OUT) |
| 2 | RA5 | Drives LED for activity indication |
| 13 | RB4 | I2C data to slaves (SDA) |
| 12 | RB5 | ASIC REPORT data input (both banks) |
| 11 | RB6 | I2C clock to slaves (SCL) |
| 10 | RB7 | ASIC REPORT clock input (both banks) |
| 16 | RC0 | ICSPDAT – programming data (otherwise unused) |
| 15 | RC1 | ICSPCLK – programming clock (otherwise unused) |
| 14 | RC2 | Analog input for Thermistor voltage |
| 7 | RC3 | DIN_9N – config data to ASIC bank 2 - P |
| 6 | RC4 | DIN_9P – config data to ASIC bank 2 - N |
| 5 | RC5 | PWM output to control fan via Power MOSFET |
| 8 | RC6 | DIN_6N – config data to ASIC bank 1 - P |
| 9 | RC7 | DIN_6P – config data to ASIC bank 1 - N |

Two Klondike board variants support either the SSOP 20-pin or QFN 20-pin packages.

Klondike Command Protocol

The Klondike device chain is controlled by a host master using a simple command protocol. Each command consists of:

<cmd> single char <address> 1 byte, followed any required data.

<address> is a single byte binary ID ranging from 0 up to <slaves>, and is always the first parameter after a command. Not shown in the table below for brevity.

Replies also start with the same <cmd><address> pattern, not shown.

| Command | Code | Data | Reply |
|----------|------|--|--|
| Identity | I | None. | <version><serial><product> |
| Status | S | None. | <state><chips><slaves><workque> <workid><temp><fanspeed> <hashcount><errorcount> |
| Config | C | <hashclock><temp target><temp critical><fan speed> or 0 binary to query only. | Updated data echoed back. |
| Work | W | <workid><midstate><data> | returns same as Status command |
| Abort | A | None. | returns same as Status command |
| Enable | E | 1 or 0 | returns same as Status command |
| Result | = | None (not sent by host). | <workid><nonce> |

The following data values are used above:

| Name | Data Type | Value |
|-------------|-----------------|---|
| <version> | 1 byte, binary | Protocol Version: high nibble:major, low nibble:minor |
| <serial> | 4 byte, binary | Unique serial# as 32bit word. |
| <product> | StringZ. | Product Identifier eg. K1, K16, K64. Zero terminated. Max 8 chars. |
| <state> | 1 character | I (initializing), R (ready for work), W (working), D (disabled). |
| <chips> | 1 byte, binary | Count of chips on board (1 - 64). |
| <slaves> | 1 byte, binary | Count of slaves connected to this board (0 – 112). |
| <workqc> | 1 byte, binary | Number of work items in queue including current active job. |
| <temp> | 1 byte, binary | Temperature value. Needs conversion to degC via calc or lookup table. |
| <fan speed> | 1 byte, binary | Fan speed. Count of ticks for tach high period. |
| <hashclock> | 2 bytes, binary | ASIC clock speed in 0.5 MHz steps, eg. 600 => 300MHz |

| | | |
|-----------------|------------------|---|
| <temp target> | 1 byte, binary | Temperature to maintain by adjusting fan speed. See temp conversion info. |
| <temp critical> | 1 byte, binary | Temperature above which to disable work. |
| <fan speed> | 1 byte, binary | Force fan speed, value 0-255 controls power to fan. |
| <workid> | 1 byte, binary | Work ID, unique per device address. |
| <midstate> | 32 bytes, binary | Midstate value to hash. |
| <data> | 12 bytes, binary | Data value to hash. |
| <hashcount> | 2 bytes, binary | Number of hashes completed as count of 256,000 hash units. |
| <errorcount> | 2 byte, binary | Number of errors that occurred in result receiver. |
| <nonce> | 4 bytes, binary | Result value of hashing. |

The provided ktest program allows manually testing commands and serves as a working example of very simple code to control the Klondike.

Preliminary Firmware Operation Details

(a more full description will follow once final development is completed)

Temperature

The value returned by protocol data is an analog device reading (ADC) not deg C. A lookup table could be used but also in capable processors a conversion using the natural log function (\ln) is easy.

From the thermistor datasheet we have,

$$\text{deg C} = 1 / (1 / (T_0 + 273.15) + \ln(R_t/R_0) / 3987) - 273.15$$

where T_0 is 25C, R_0 is 2200, R_t is the thermistor resistance measured via the voltage divider network and given by

$R_t = \text{ADC} * 1000 / (256 - \text{ADC})$ and is independent of reference voltage.

Or, in C code from cgminer klondike driver:

```
return (double)1/((double)1/(25+273.15) +  
    log((double)temp*1000/(256-temp)/2200)/3987) - 273.15;
```

Note that the temperature is read from a thermistor near the U7 ASIC and so is not the internal temperature of the ASIC. An offset value should be applied that estimates the difference. This offset is undetermined at this time.

See the “Thermistor Lookup Table.ods” spreadsheet for example lookup and temperature conversion details.

ASIC Clock

ASIC clock speed is indicated in 0.5 MHz steps – double the MHz rate, eg.

282 → write 564

300 → write 600

Fan Speed

Fan tachometer reading are the measured high period in 170uS ticks. Since the maximum is

255 ticks that means a high period of 43,350 uS. Since there are two tachometer periods per RPM with 50% duty cycle that makes the lowest possible RPM reading,

$$1/(0.043350 * 2 * 2) * 60 \text{ secs/minute} = 346 \text{ RPM.}$$

Below this it should report zero as a rollover clears the count.

Hash Timer

The code uses Timer0 as a hash timer to predict when new work should be pushed and time how much work was done if aborted.

Timer0 runs off the instruction clock at 12 MHz with a DIV 256 prescaler, making each count 21.3 uS. A count is loaded into Timer0 depending on ASIC hash clock speed such that each rollover occurs at close to 256,000 hash cycles. This makes counting rollovers as “ticks” useful for counting large hash values to a resolution of 256,000 – and potentially more accurate if the Timer0 value is checked, though that is not currently implemented.

For example, at 282MHz clock a value of (256-42) is set for Timer0 which causes a tick every 896 uS. This corresponds closely with $282,000,000 / 256,000$ hashes completed (actually 252,672 hashes). This tick is counted to determine when new work should be pushed to the ASIC chain. The work tick value is dependent on how many ASICs are populated on each board.

The work tick value is also counted up to 256 to provide a “Slow Tick” for various housekeeping tasks such as updating fan speed, temperature and toggling the activity LED.