Bi*fury USB communication protocol, protocol version 2.

Arlet Ottens, c-scape <arlet@c-scape.nl>

USB properties

The bi*fury USB device has VID=0x198C and PID=0xB1F1. It presents itself as a CDC function device, but it can also be used directly using bulk endpoints 2 IN/OUTs. The device implements a virtual serial device over these bulk endpoints. All commands and responses are newline (ASCII 10) terminated lines of plain text.

Commands

There are two core commands from the host to the bi*fury device, 'work' and 'target'.

```
work  <work string> <job id>
```

The work string is a hex string, which should be the same as the first 76 bytes of the getwork protocol string (nonce and padding removed). The job id is a 32 bit hexadecimal number that the host should provide to identify the work item. The bi*fury firmware will extract the prevhash and timestamp from the work string. If the prevhash has changed, the old work will be cancelled as quickly as possible.

The firmware will start looking for nonces at the supplied timestamp, and will automatically increase the timestamp by steps of +1 every time the nonce search space has been exhausted. The maximum time increment can be set with the 'maxroll' command. Alternatively, the host can check the timestamp in the responses to see whether new work is required. A minimum of one work item is needed when maxroll is not used (or set to a high value). Note that the firmware will roll the timestamp beyond the maximum specified in maxroll if the host does not send work quickly enough.

Note: hex strings are not prefixed by '0x'. Firmware only produces lower case a-f, but will also accept upper case.

```
target  <target number>
```

The 'target' command tells the firmware the desired difficulty of returned blocks. The target number is a 32 bit hexadecimal number, which gives the upper limit of the hash. The firmware compares the target value with the 7th word of the hash (the 8th word must be zero).

If the target number equals FFFFFFFF, the firmware returns difficulty 1 shares. If the target value equals 7FFFFFFE, the firmware only returns difficulty 2 shares. For any given difficulty, target should be FFFFFFFF / difficulty.

There are also a few non-essential commands:

flush

The 'flush' command invalidates all old work. To continue mining, the host needs to send new 'work' items as quickly as possible.

maxroll *<seconds>*

The 'maxroll' command set the number of seconds that the time may be advanced by the device. The argument is a decimal number. If the number is set to 0, no time rolling is done. When the device first initializes, a very high value is used for maxroll, effectively disabling any time limit.

version

This command requests to get a 'version' response.

clock *<clock0> ><clock1>*

This command allows the clock speed of the Bitfury chips to be set. The arguments are decimal numbers between 0 and 63. A value between 0 and 32 turns the chip off. A value between 32 and 63 turns the chip on, with increasing clock rate for higher numbers. The reset default is a value of 54, which typically results in the best performance. If the value is too high, the chips will start to make errors, resulting in a drop of effective hashing rate.

Responses

Note: The responses are all asynchronous. So, for instance, if the 'version' command is sent, the firmware will send a version response, but there may be some 'temp' or 'submit' responses first. The host should not sit and wait for a particular response, but be prepared to handle everything that comes in, in any arbitrary order.

The bi*fury firmware has one core response:

submit *<nonce> <job id> <timestamp> <chip>*

The first 3 values are 32 bit hexadecimal values. The nonce is the found nonce (same as in getwork protocol), the job id is the same number as was supplied in the 'work' command, and the timestamp is the same timestamp as supplied in the work + an offset determined by the bi*fury. The chip number (starting at 0) responsible for finding the nonce is given as well.

The firmware also periodically reports the board temperature in integer units of 0.1 degrees Celsius

temp *<temperature>*

For example, 'temp 631' means a temperature of 63.1 degrees.

needwork *<items>*

This response indicates how many work items the firmware requires to run smoothly. The 'items' argument is a decimal number corresponding to the number of 'work' lines the host should send (as quickly as possible). This is an optional response from the firmware: if the firmware has plenty of work queued (especially when maxroll isn't used, or set to a high value), it won't use this response.

version *<major>.<minor> rev <hw rev> chips <number of chips>*

Shows the current version number, hardware revision and number of chips. All arguments are decimal numbers.

job *<job id> <timestamp> <chip>*

This response indicates when a chip is done processing a job + timestamp combination. This could be used for diagnostics/statistics. Even when a chip does not submit valid shares, it should still produce 'job' responses. A lack of job responses could indicate a chip is disabled or broken. Also, the ratio between job/submit can be used to calculate the chip error rate. On a perfect chip the ratio should be 756/1024 between submit shares and job responses. In addition, the job timing could be used to calculate max theoretical hashing speed.

*Example session (commands from host in black, responses from bi*fury in blue)*

```
target 7fffffff

work
0000000260f77244da41f7ca9e1085820f7b493e75159a1e0e7cccb40000000d00000000c60bac09
7b17d4d2be6966efe34d215b71da333f6c2708f05fcac9d13cbceb63525e6a7f1916b0ca 3

work
0000000260f77244da41f7ca9e1085820f7b493e75159a1e0e7cccb40000000d00000000825d9b97
1d8b991beab7fd43ce6cc049846c4f27e758fb25a7e0912ee780907f525e6a7f1916b0ca 4

submit a2c1b5b6 3 525e6a80

submit 599eefe1 3 525e6a7f

submit 29beee47 3 525e6a7f

submit f2472c8a 4 525e6a81
```

Extra Commands/Responses

Newer versions may add extra commands or responses. To ensure maximum compatibility, firmware and host should silently ignore any command or response they don't understand.