We claim:

1. A method for stack timing adjustment for serial communications, the method comprising:

receiving a USB communication from a USB device configured to encode a serial packet into the USB communication;

extracting and decoding the serial packet from the USB communication; and

adjusting a timing of the serial packet according to a serial protocol.

2. The method of claim 1, wherein the step of extracting and decoding the serial packet from the USB communication comprises ordering UART frames.

3. The method of one of claim 1 or claim 2, wherein the step of adjusting the timing of the serial packet according to a serial protocol comprises adding a delay between two or more serial packets encoded into the USB communication.

4. The method of one of any of the foregoing claims, wherein the step of adjusting the timing of the serial packet according to a serial protocol comprises removing inter-character delays added to the serial packet encoded into the USB communication.

5. The method of one of any of the foregoing claims, wherein the serial protocol comprises a Modbus protocol.

6. The method of one of any of the foregoing claims, wherein the serial protocol comprises a HART protocol.

7. The method of one of any of the foregoing claims, wherein the USB communication comprises a USB CDC message with a serial packet.

8. A USB device with stack timing adjustment (100) for serial communications, comprising:

a USB controller (100a) configured to:

receive a USB communication from a USB device configured to encode a serial packet into the USB communication; and

extract the encoded serial packet from the USB communication;

a microprocessor (100b) configured to decode the encoded serial packet; and

a stack timing adjustment (100c) configured to adjust a timing of the serial packet according to a serial protocol.

9. The USB device with stack timing adjustment (100) according to claim 8, wherein the microprocessor (100b) is further configured to order UART frames in a sequence.

10. The USB device with stack timing adjustment (100) of one of claim 8 or claim 9, wherein the stack timing adjustment (100c) configured to adjust the timing of the serial packet according to the serial protocol comprises the stack timing adjustment (100c) configured to add a time delay between two or more of the serial packets encoded into the USB communication.

11. The USB device with stack timing adjustment (100) of one of any of the foregoing claims 8 through 10, wherein the stack timing adjustment (100c) configured to adjust the timing of the serial packet according to the serial protocol comprises the stack timing adjustment (100c) configured to remove inter-character delays added to the serial packet encoded into the USB communication.

12. The USB device with stack timing adjustment (100) of one of any of the foregoing claims 8 through 11, wherein the serial protocol comprises a Modbus protocol.

13. The USB device with stack timing adjustment (100) of one of any of the foregoing claims 8 through 12, wherein the serial protocol comprises a HART protocol.

14. The USB device with stack timing adjustment (100) of one of any of the foregoing claims 8 through 13, wherein the USB communication comprises a USB CDC message with a serial packet.

15. A communications system with stack timing adjustment (50) for serial communications, comprising:

a USB device (200) that is adapted to encode a serial packet into a USB communication; and

a USB device with stack timing adjustment (100) in communication with the USB device (200) and configured to:

extract and decode the serial packet from the USB communication; and

adjust a timing of the serial packet.

16. The communications system with stack timing adjustment (50) of claim 15, wherein the USB device with stack timing adjustment (100) is further configured to respond to the serial packet with a serial response packet that is encoded into a response USB communication.

17. The communications system with stack timing adjustment (50) of one of claim 15 or claim 16, wherein the USB device (200) is a master that is further configured to encode a serial request packet into the USB communication.