User Name Password Log in Help Register

□ Remember Me?

What's New? Forum

New Posts FAQ Calendar Community Forum Actions Quick Links Advanced Search

Forum Main Category Technical Support & Questions USB serial data loss

Forum Rule: Always post complete source code & details to reproduce any issue!

If this is your first visit, be sure to check out the **FAQ** by clicking the link above. You may have to **register** before you can post: click the register link above to proceed. To start viewing messages, select the forum that you want to visit from the selection below.

Results 1 to 5 of 5

Thread: USB serial data loss

Thread Tools Search Thread Display

11-20-2018, 09:20 PM #1

markb

Member

Join Date: Dec 2016

Posts: 42

USB serial data loss

I'm using a Teensy 3.2 with Teensyduino 1.44.

I have an application where I send data from a PC to the Teensy over the USB serial port, and I was noticing data corruption. I wrote this simple sketch to demonstrate the problem:

Code:

```
void setup() {
  Serial.begin(1000000);
}

static int seq = 0;
static int received = 0;
static int lasterror = -1;

void loop() {
  if (Serial.available()) {
    int c = Serial.read();
    if (c != seq) {
      Serial.print("Received: ");
      Serial.println(c);
      Serial.print("Expected: ");
      Serial.println(seq);
      if (lasterror >= 0) {
        Serial.print("# bytes since last error
        Serial.println(received - lasterror);
      }
      lasterror = received;
    }
    seq = c + 1;
    if (seq > 255) {
      seq = 0;
    }
    received++;
    if (received % 150000 == 0) {
      Serial.print("# of bytes received: ");
      Serial.println(received);
```

On the PC side, I'm running this code (C#):

Code:

```
using System;
using System.IO.Ports;
using System.Threading;

namespace teensy_usbserial
{
    class Program
    {
        private static void DataReceivedHandle
        {
            var serial = (SerialPort)sender;
            Console.Write(serial.ReadExisting(
        }

        static void Main(string[] args)
        {
            var serial = new SerialPort("COM11
            serial.DataReceived += new SerialDa
            serial.Open();
            var buf = new byte[1572];
            byte seq = 0;

            while (true) {
                for (int i = 0; i < buf.Length
                    buf[i] = seq++;
                }
                serial.Write(buf, 0, buf.Lengt
                Thread.Sleep(200);
            }
        }
```

And, finally, the output (from the PC, which just shows what the Teensy sends to it):

Code:

```
Received: 172
Expected: 44
Received: 80
Expected: 208
# bytes since last error: 36
Received: 236
Expected: 172
# bytes since last error: 92
Received: 208
Expected: 16
# bytes since last error: 36
Received: 52
Expected: 180
# bytes since last error: 78308
Received: 244
Expected: 116
# bytes since last error: 64
Received: 116
Expected: 52
# bytes since last error: 64
Received: 116
Expected: 180
# bytes since last error: 64
Received: 88
Expected: 152
# bytes since last error: 64
Received: 12
Expected: 140
# bytes since last error: 7988
Received: 204
```

These numbers suggest that I'm loosing entire USB packets, because most of those packets should be carrying 64 bytes of data. Is there just too much overhead in the loop to keep up? It does very little work, unless it finds an error.

Reply With Quote

11-20-2018, 11:26 PM #2

markb ○

Member

Join Date: Dec 2016
Posts: 42

After some more investigation, I found that changing "if (Serial.available()) {" to "while (Serial.available()) {" solved the problem! So this suggests that there is too much time between loop() calls to keep up with the USB stream. I was even able to reduce the delay in the sending code all the down to 1ms, and still no data loss.

So, I think the lesson here is don't return control from loop() until you've processed all the available data. Returning from loop() occasionally looks necessary for certain functionality (events), though, so some combination of the two strategies might be best.

Reply With Quote

11-21-2018, 08:30 PM #3

Frank B ○

Senior Member+

Join Date: Apr 2014
Location: Germany
Posts: 7,936

There is no reason to call yield() (by returning from loop()) if you don't need the events. And even if you need the events..take a look at the sourcecode of yield() - it is so simple...
yield() checks _all_ UARTS. I guess not more than 0.01% of Arduino users need _all_ events for_all_uarts, but it slows down _all_ sketches... well..Arduino design 😊
Luckyly, the solution is simple - just make your own loop.

For your problem - dataloss - perhaps increasing the buffer sizes helps. There may be a simple buffer overflow*? (<- I mean, not enough buffers available)

Last edited by Frank B; 11-21-2018 at 09:02 PM.

Reply With Quote

11-22-2018, 02:24 AM #4

markb ○

Member

Join Date: Dec 2016
Posts: 42

> Originally Posted by **Frank B**
>
> There is no reason to call yield() (by returning from loop()) if you don't need the events. And even if you need the events..take a look at the sourcecode of yield() - it is so simple...
> yield() checks _all_ UARTS. I guess not more than 0.01% of Arduino users need _all_ events for_all_uarts, but it slows down _all_ sketches... well..Arduino design 😊
> Luckyly, the solution is simple - just make your own loop.
>
> For your problem - dataloss - perhaps increasing the buffer sizes helps. There may be a simple buffer overflow*? (<- I mean, not enough buffers available)

I did take a look at the yield() source code, but I didn't follow it all the down. There's a call to EventResponder::runFromYield(). Not really sure what that does, but it's probably nothing important for my project.

Regarding buffer size, what buffer are you talking about?

As you can see from the Teensy source code I posted above, I didn't create any buffers. Is there a way to have Teensy buffer more USB packets?

Reply With Quote

11-22-2018, 07:20 AM #5

Frank B

Senior Member+

Join Date:	Apr 2014
Location:	Germany
Posts:	7,936

> Originally Posted by **markb**
>
> I did take a look at the yield() source code, but I didn't follow it all the down. There's a call to EventResponder::runFromYield(). Not really sure what that does, but it's probably nothing important for my project.
>
> Regarding buffer size, what buffer are you talking about? As you can see from the Teensy source code I posted above, I didn't create any buffers. Is there a way to have Teensy buffer more USB packets?

Oh, I had forgotten (I haven't been here for a long time) that all this was replaced by the eventresponder, which now runs with every systick+yield. Sorry for that!..
If you scroll down here: https://www.pjrc.com/teensy/td_serial.html You'll see some useful tips. For setting the buffersize, you may want to look at the code.

Reply With Quote

Quick Navigation **Technical Support & Questions** **Top**

« Previous Thread | Next Thread »

Posting Permissions

You may not post new threads
You may not post replies
You may not post attachments
You may not edit your posts

BB code is On
Smilies are On
[IMG] code is On
[VIDEO] code is On
HTML code is Off

Forum Rules

-- Default Style

PJRC Electronic Projects Archive Top

All times are GMT. The time now is 10:47 AM.

Powered by vBulletin® Version 4.2.2
Copyright © 2021 vBulletin Solutions, Inc. All rights reserved.
Web Hosting