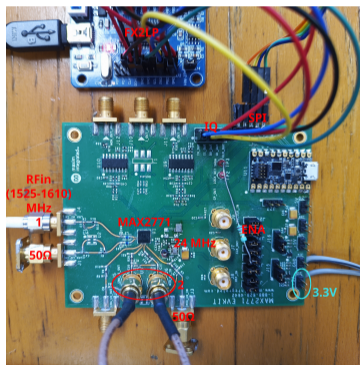


Efficient USB communication under GNU/Linux for a wideband (MAX2771-based) L-band (GNSS) SDR receiver

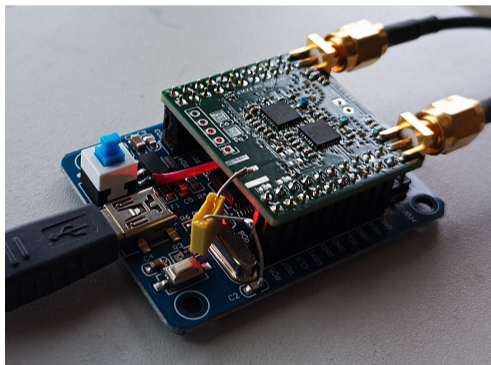
J.-M Friedt

FEMTO-ST Time & Frequency, Besançon, France

From

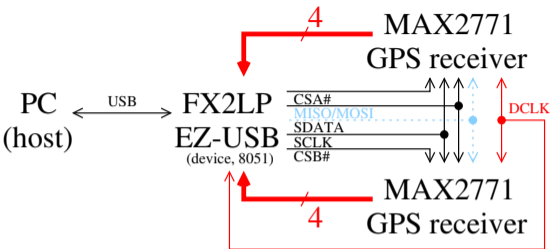


to



October 19, 2024

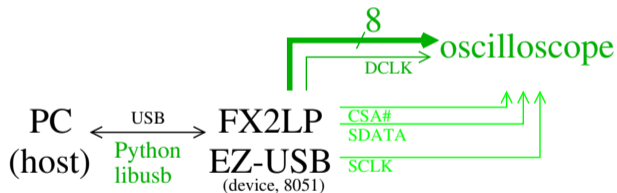
High bandwidth transfer



- ▶ Bulk USB transfer: from device to host and from host to device
- ▶ Hardware pins determine the data flow direction
- ▶ Registers determine the configuration of USB endpoints
- ▶ Callback functions (interrupt service routines) handle USB events

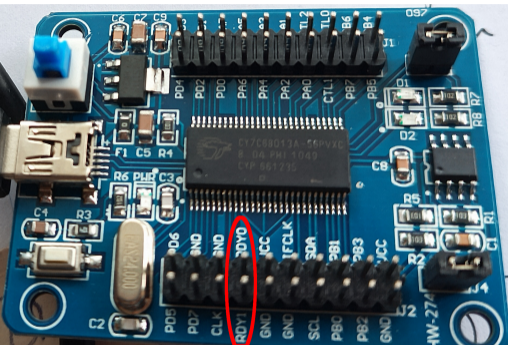
Proposed objectives

- ▶ show how you can generate a pattern on the FX2LP output port by sending associated messages through bulk transfer from host to device
- ▶ see how you can read from device to host



- ▶ output from one FX2LP and read from another?

FX2LP board and software



Tomoji Takasu ^a: RDY0 and RDY1 defining communication direction are swapped

^a<https://blog.goo.ne.jp/osqzss/e/d86df04de96123fd5c73bbb6db6e8bc5>

USB programming: Vendor requests

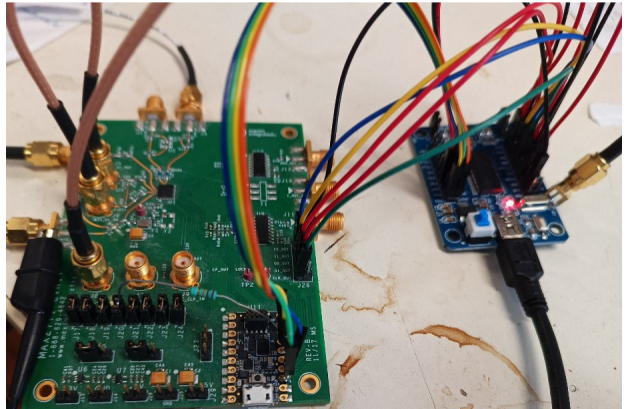
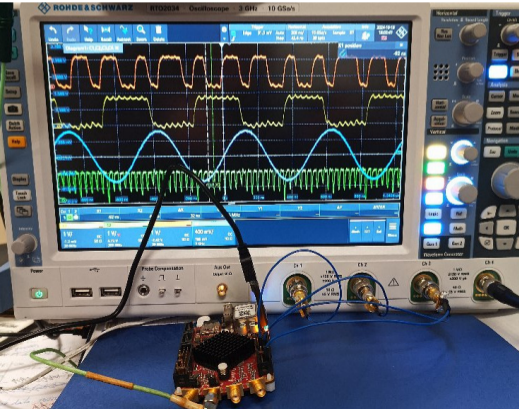
```
BOOL handle_vendorcommand(BYTE cmd) {
    uint16_t len = WORD_(SETUPDAT + 6);
    uint16_t val = WORD_(SETUPDAT + 2);
    uint32_t val32;
    uint8_t ctrl;
    switch ( cmd ) {
        case VR_REG_READ:
            { val32=read_reg(SETUPDAT[3], SETUPDAT[2]);
              *(uint32_t *)EP0BUF = val32;
              EP0BCH = 0;
              EP0BCL = 4;
              return TRUE;
            }
            break;
    }
    ...
}
```

Bulk transfers

```
REVCTL = bmNOAUTOARM | bmSKIPCOMMIT; // REVCTL = 0x03;
EP6CFG = 0xe2; // 1110 0010 (bulk IN, 512 bytes, double-buffered)
FIFORESET = 0x80; // NAK all requests from host.
FIFORESET = 0x82; // Reset individual EP (2,4,6,8)
FIFORESET = 0x84;
FIFORESET = 0x86;
FIFORESET = 0x88;
FIFORESET = 0x00; // Resume normal operation.
while (1)
{
    if (!(EP2468STAT & bmEP6FULL)) // Wait for EP6 buffer to become non-full
    {
        for (int i = 0; i < 512; i += 2)
        {
            EP6FIFOBUF[i] = IOB; // fill buffer with port b and d
            EP6FIFOBUF[i + 1] = IOD;
        }
    }
}
```

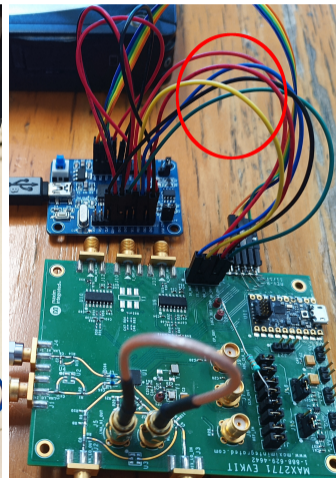
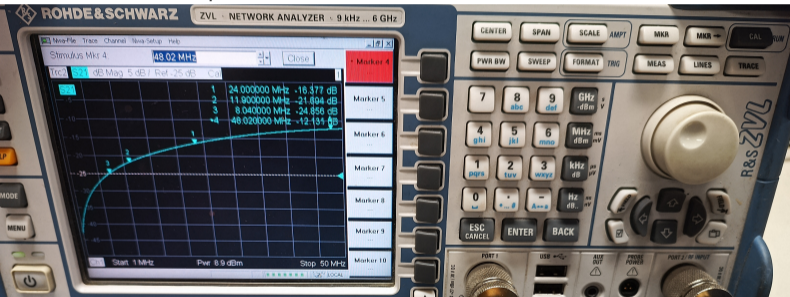
Resources

- ▶ `fx2lib/examples/bulkloop/bulkloop.c` and its `BOOL handle_vendorcommand(BYTE cmd) {}` implementation
- ▶ USB device template at `fx2lib/fw`
- ▶ how is the link between interrupt service routine in `fx2lib/lib/interrupts` and `handle_vendorcommand()` achieved?
- ▶ probing fast signals: reduce cable length and avoid probes. Will route PCB to avoid crosstalk.



Resources

- ▶ probing fast signals: reduce cable length and avoid probes. Will route PCB to avoid crosstalk.
 - ▶ Matched transmission line maximizes current and hence cross-talk (coupled inductance/capacitance)
 - ▶ Unmatched transmission line minimizes current but induces ringing and reflections
 - ▶ X MHz square wave extends to $N \times X$ MHz, $N \in \mathbb{N}$



08 MHz : -24,9 dB = 0,003 / -45 dB
12 MHz : -21,7 dB = 0,007 / -41 dB
24 MHz : -16,4 dB = 0,023 / -33 dB
48 MHz : -12,1 dB = 0,062 / -28 dB

