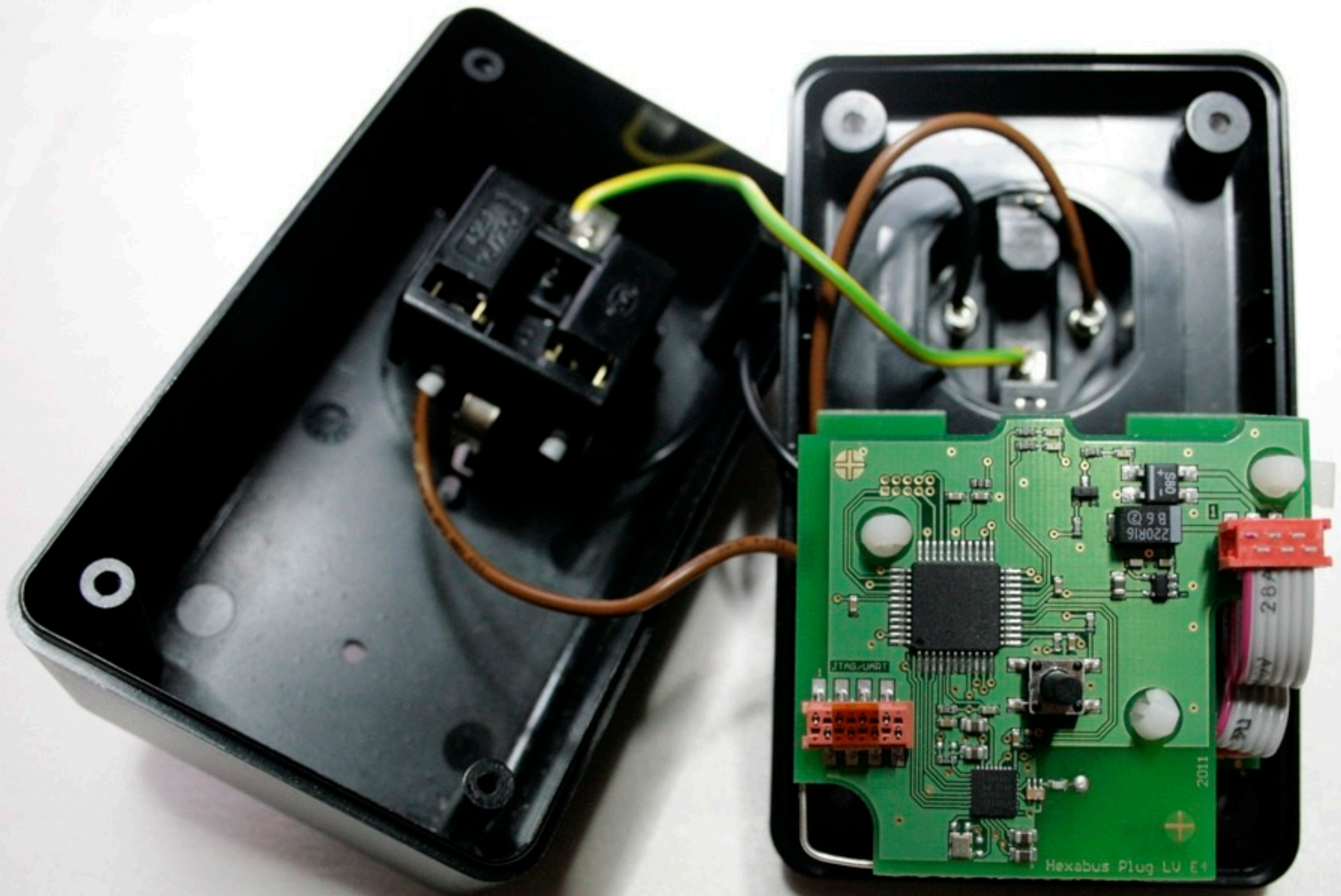
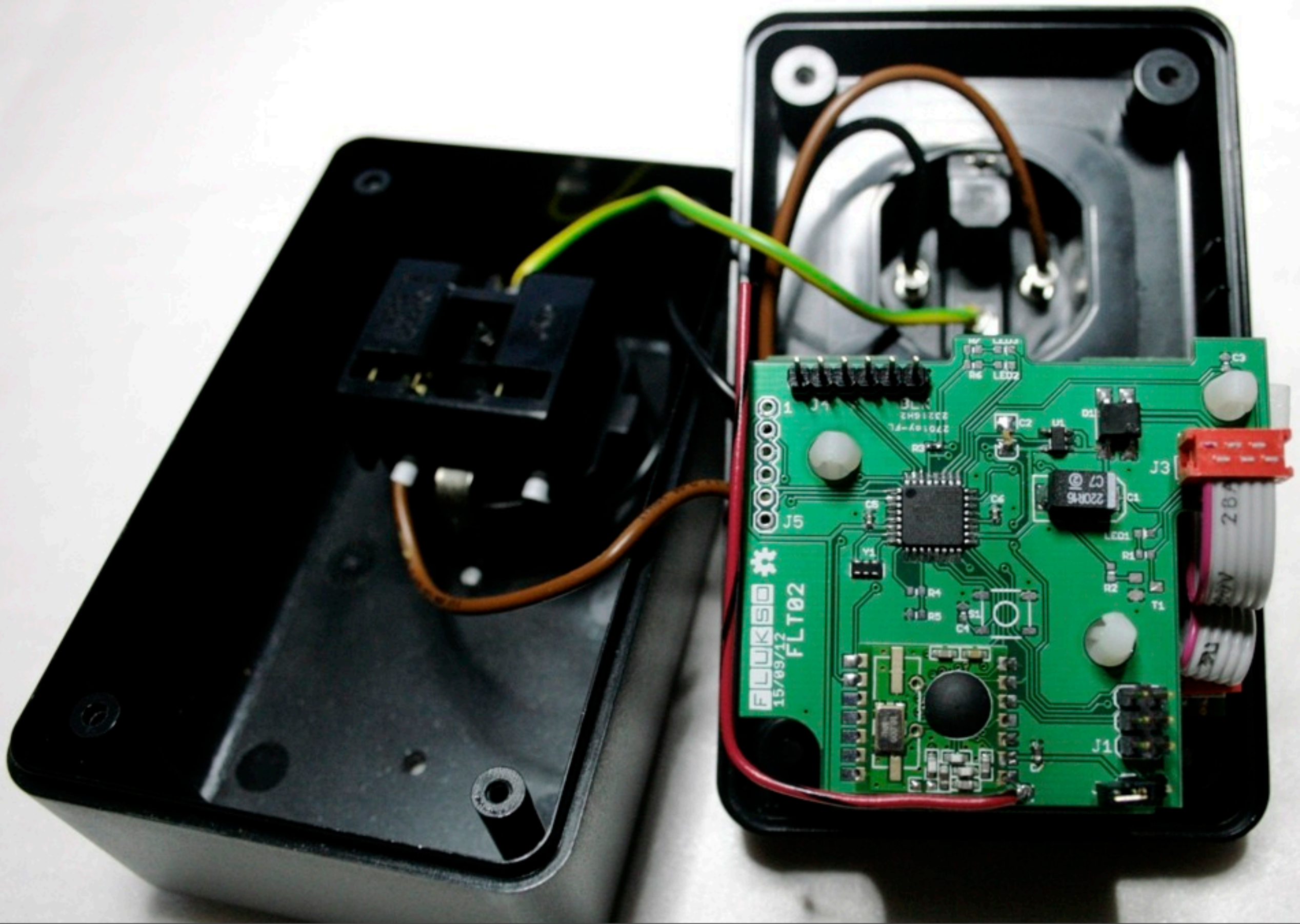




community metering





Access to ADE7768 and relay

FTDI header

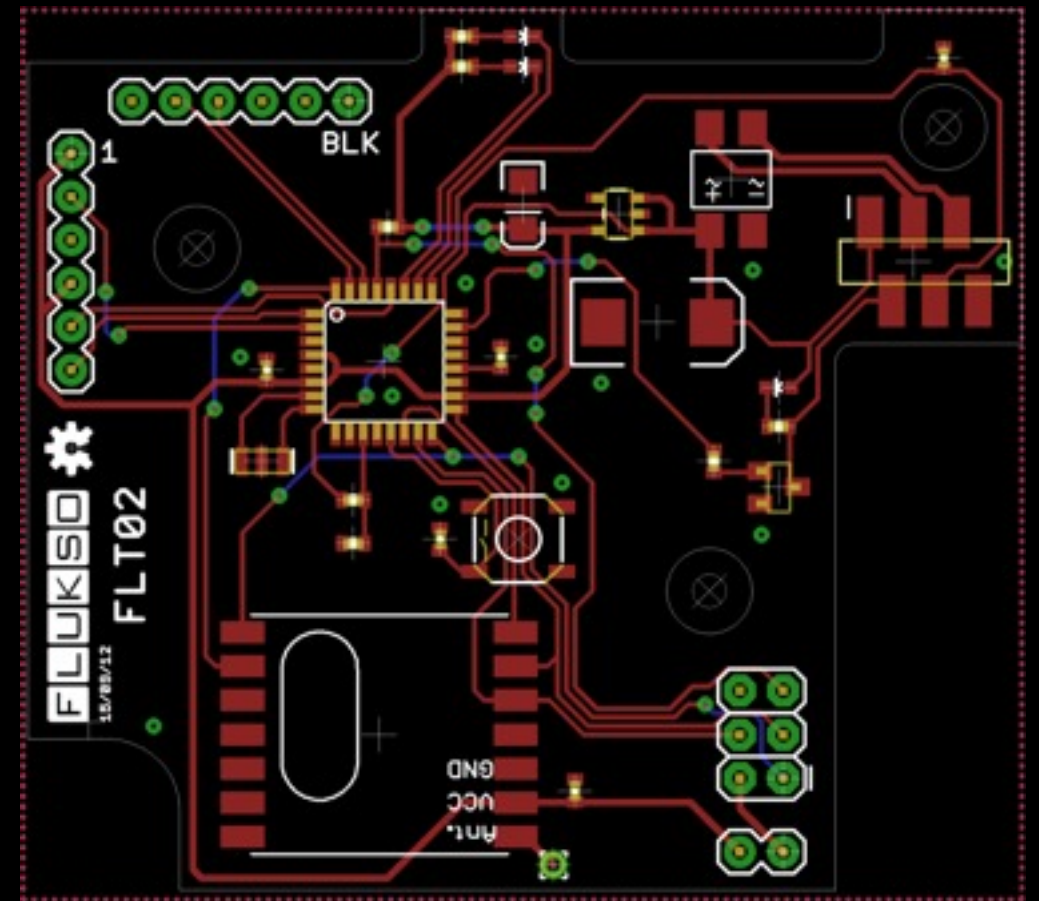
ATmega328p

Jeenode 'port'

RFM12B

ICSP

- Hw & Sw sources available in Flukso repo
- <https://github.com/flukso/flt02>
- Should be Arduino/Jeelib compatible
- Extend the basic setup with Jeeplugs
- Have a multi-functional always-on 'room node'




```

static pkt_1_t pkt_counter = {
    .head = {
        .grp = PKT_GRP, //hardcoded for the time being
        .hdr = PKT_NID, //CTL=0, DST=0, ACK=0 -> b'cast noack
        .len = PKT_1_LEN,
        .typ = PKT_1_TYP
    },

    .sid = 0,
    .count = 0,
    .msec = 0
};

...

int main(void)
{
    cli();
    _delay_ms(10);

    rfm12_init();
    timer0_init();
    s0_init();

    // the clk/8 fuse bit is set
    clock_prescale_set(clock_div_1);
    sei();

    while (1) {
        if (pkt_counter_send && !send((uint8_t *) &pkt_counter, sizeof(pkt_counter))) {
            //clear when pkt is in tx buffer
            //if not, the tx buffer is still occupied, so try again later
            pkt_counter_send = false;
        }

        rfm12_tick();
    }
}

```

```

#define PKT_GRP          0xd4
#define PKT_NID          0x01

typedef struct {
    uint8_t grp;
    uint8_t hdr;
    uint8_t len;
    uint8_t typ;
} pkt_head_t;

#define PKT_1_LEN10
#define PKT_1_TYP1

typedef struct {
    pkt_head_t head;
    uint8_t sid; //local sensor id
    uint32_t count; //in mWh
    uint32_t msec;
} pkt_1_t;

```

- Double function
 - Keeping sensor/actuator state
 - Packet formatting
- Casting the type as a byte array will result (for AVR) in little-endian encoding on the wire
- MIPS is big-endian
- Can we use the packet typedef for correct decoding on the receiver side?
- Can we define a set of generic packet types?

- repo: <https://github.com/flukso>
- forum: www.flukso.net/forum
- dev mailing list: flukso-dev-join@lists.flukso.net
- freenode: #flukso







Macarena C., ?

www.flickr.com/photos/room_onfire/403830495/