```
typedef struct {
  uint16_t v1; // Voltage for first preference - mapping in read_config()
  uint8_t i1; // Current for first preference - mapping in current lookup table of STUSB4500 datasheet
  uint16_t v2; // Voltage for second preference - mapping in read_config()
  uint8_t i2; // Current for second preference - mapping in current lookup table of STUSB4500 datasheet
  bool req_pd; // If true, requires source to be PD capable before enabling output
} stusb4500_config;

// read_config blocks until a successful configuration is read from the
// light sensor and returns it.
stusb4500_config read_config() {
  uint16_t voltage_map[] = {5 * 20, 9 * 20, 12 * 20, 15 * 20, 20 * 20};
  enum { packet_len = 4 };
  uint8_t packet[packet_len];
  static uint16_t ptp_buffer[packet_len * 8];
  static uint8_t ptp_next_bit_pos;
  while (1) {
    read_packet(packet, ptp_buffer, &ptp_next_bit_pos, packet_len);
    if (packet[0] & 0b10000000) {
      continue;
    }
    stusb4500_config c;
    c.v1 = (packet[0] >> 4) & 0b111;
    c.v2 = (packet[1] >> 4) & 0b111;
    if (c.v1 > 4 || c.v2 > 4) {
      continue;
    }
    c.v1 = voltage_map[c.v1];
    c.v2 = voltage_map[c.v2];
    c.i1 = packet[0] & 0xf;
    c.i2 = packet[1] & 0xf;
    if (c.i1 == 0 || c.i2 == 0) {
      continue;
    }
    c.req_pd = packet[1] >> 7;
    return c;
  }
} // read_config
```