

LibParticle

LibParticle

Homepage: <http://particle.teco.edu>

Benutzt „scons“ als Buildsystem

Fürs SEP den Befehl „scons lang=none“ dadurch wird nur die C – Library erstellt.

LibParticle

Von LibParticle zur Verfügung gestellte funktionen

- socket - die Packete Senden und Empfangen
- filter - Packete sehr früh ausfiltern
- packet - Packete Erstellen, Manipulieren und Auswerten
- acl - Die Daten der Packete
- db - Datenbank anbindung

LibParticle - packet

```
struct p_packet
{
    /** 1 byte for the flags of the packet */
    int    p_flags;
    /** destination of the packet
     * @sa: p_socket_set_ipdest
     */
    uint8_t p_dest_id[8];
    /** acking sequence-number */
    uint8_t p_ack_seq;

    struct sockaddr p_bgd_peer;

    socklen_t    p_bgd_len;

    /** CL HEADER START */

    /** Version of the CL-Packet */
    uint8_t p_version;
    /** location ID (44 Byte) */
    uint8_t p_location[44];
    /** Source ID (8 Byte) */
    uint8_t p_src_id[8];
    /** Sequence number */
    uint8_t p_seq;

    /** CL HEADER END */

    /** size of the ACL-payload */
    size_t  p_payload_size;

    /** counter for the ACL-tuples */
    size_t  p_acl_count;
    /** list of all the ACL-tuples */
    struct p_acl_list p_acl_list;
    /** pointer to the first ACL-tuple */

    TAILQ_ENTRY(p_packet) entries;
};
```

Packete:

Bestehen aus einem Header mit grundlegenden Informationen und einer verketteten liste aus ACL – Tupeln - den eigentlichen Daten.

Wegen der ACL – Liste muss `p_pkt_free(packetname)` verwendet werden, um ein paket zu löschen.

LibParticle - acl

ACL – Tupel bestehen aus zwei Teilen:

acl-type - 3 Buchstaben, in 2 byte dargestellt (p_util_3to2())
Definieren den Typ, z.B.
S LI : Sensordaten Licht
S TE : Sensordaten Temperatur

<http://particles.teco.edu/documentation/content/aclh.html>

acl-data – die eigentlichen Daten in form eines uint8

LibParticle - filter

Die libParticle hat bereits ein System, um Pakete zu Filtern.

```
int filter_nr = p_filter_create("beliebiger_filter_name");  
p_filter_add(filter_nr, FILTER_TYPE_*|FILTER_CONCAT_*|FILTER_MODE_*, pattern);
```

```
FILTER_CONCAT_AND  
FILTER_CONCAT_OR
```

```
FILTER_TYPE_LOCATION  
FILTER_TYPE_ID  
FILTER_TYPE_ACL,  
FILTER_TYPE_ACL_POSITION  
FILTER_TYPE_FILTER
```

Der erste filter MUSS FILTER_CONCAT_AND sein.

```
FILTER_MODE_WILDCARD  
FILTER_MODE_NOT
```

```
ing filter3 = p_filter_create("beispiel_f3");  
p_filter_add(filter3, FILTER_TYPE_ID|FILTER_CONCAT_AND, "1.2.3.4.0.4.0.177");  
p_filter_add(filter3, FILTER_TYPE_ID|FILTER_CONCAT_OR, "1.2.3.4.0.4.0.83");  
p_pkt_set_srcid(testpaket, "1.2.3.4.0.4.0.83");  
assert (p_filter_apply(filter3, testpaket) == 1);  
p_filter_destroy(filter3);
```