

```

1  /*
2  Serial packet handler library
3
4  Author: Heethesh Vhavle
5  Version: 1.0.0
6  Email ID: heethesh@cmu.edu
7  */
8
9  #include "communication.h"
10
11  TXDataPacket tx_packet;
12  RXDataPacket rx_packet;
13
14  void send_data()
15  {
16      // Packet metadata
17      uint8_t size = sizeof(TXDataPacket);
18      uint8_t checksum = 0;
19
20      // Payload buffer
21      uint8_t buff[size];
22
23      // Convert struct to bytes
24      memcpy(buff, &tx_packet, size);
25
26      // Send header
27      Serial.print("$>");
28
29      // Send data length
30      Serial.write(size);
31
32      // Send byte array and update checksum
33      for (int i = 0; i < size; i++) {
34          Serial.write(buff[i]);
35          checksum ^= buff[i];
36      }
37
38      // Send checksum
39      Serial.write(checksum);
40  }
41
42  bool recieve_data()
43  {
44      // Packet metadata
45      uint8_t size = sizeof(RXDataPacket);
46      uint8_t checksum = 0;
47
48      // Payload buffer
49      uint8_t buff[size];
50
51      // Check (frame + header) length
52      if (Serial.available() >= size + 4) {
53          // Check header of data frame
54          if (Serial.read() != '$') {
55              return;
56          }
57          if (Serial.read() != '<') {
58              return;
59          }
60
61          // Data length byte
62          Serial.read();
63
64          // Read data bytes

```

```
65     for (int i = 0; i < size; i++) {
66         buff[i] = Serial.read();
67         checksum ^= buff[i];
68     }
69
70     // Discard frame if checksum does not match
71     if (Serial.read() != checksum) {
72         return;
73     }
74
75     // Convert bytes to struct
76     memcpy(&rx_packet, buff, size);
77
78     // Success
79     return true;
80 }
81
82 // Failure
83 return false;
84 }
```