

Не защищено | 192.168.14.222/card.html

Подтвердите действие на странице 192.168.14.222

<Access Cards>
is completed. <3002> out of <3002>. Successfully

OK

Card Number	PIN	RDR#1 ACCSSLVL	RDR#2 ACCSSLVL	RDR#3 ACCSSLVL	RDR#4 ACCSSLVL
+ ...					

HTTP-KEYPAD-WU (EM2000W) RUN - TIDE - main

```

65 for (i=0;i<3;i++)
66 {
67     sock.num=i;
68     sock.rxbuffrq(1);
69     sock.txbuffrq(1);
70     sock.redir(PL_REDIR_SOCK# + s
71     sys.buffaloc();
72     sock.protocol=PL_SOCK_PROTOCD
73 }
  
```

Stack size: 30 bytes
Build complete.
Loading program database... Complete.

Uploading...
Application file successfully loaded!
Debug session started...

643 requests | 352 KB transferred | 367 KB resources | Refresh: 6.0 min | DOMContentLoaded: 4.09 s | Load: 13.56 s

Uncaught (in promise) Error: The message port closed before a response was received. at contest.js:25

Uncaught (in promise) Error: the message port closed before a response was received. at contest.js:25

In my case, the instructions on https://docs.tibbo.com/taiko/sock_http_buff regarding performance are true, sock.txbuffrq (4) allows you to send 255 KB in 11 seconds.

Не защищено | 192.168.14.222/card.html

Подтвердите действие на странице 192.168.14.222

<Access Cards>
is completed. <3002> out of <3002>. Successfully

OK

Card Number	PIN	RDR#1 ACCSSLVL	RDR#2 ACCSSLVL	RDR#3 ACCSSLVL	RDR#4 ACCSSLVL
+ ...					

HTTP-KEYPAD-WU (EM2000W) RUN - TIDE - main

```

65 for (i=0;i<3;i++)
66 {
67     sock.num=i;
68     sock.rxbuffrq(1);
69     sock.txbuffrq(4);
70     sock.varbuffrq(4);
71     sock.redir(PL_REDIR_SOCK# + s
72     sys.buffaloc();
73     sock.protocol=PL_SOCK_PROTOCD
  
```

Stack size: 30 bytes
Build complete.
Loading program database... Complete.

Uploading...
Application file successfully loaded!
Debug session started...

465 requests | 301 KB transferred | 367 KB resources | Refresh: 4.4 min | DOMContentLoaded: 1.06 s | Load: 1.83 s

Uncaught (in promise) Error: The message port closed before a response was received. at contest.js:25

Uncaught (in promise) Error: the message port closed before a response was received. at contest.js:25

I have something to work on. Creating the table itself is slow. Probably the loop that calculates the insertion point of each added row, and the editable elements with a set of functions in each field interfere with me. Your sample lacks such functionality, so in your case the table is built quickly and blocking is often invisible.