```
//Sidus F1Q/B/E - Commands

#define CMD_GETTIMERS       'a'          // request timers
#define RSP_GETTIMERS       'b'          // reply to request timers
#define CMD_SETSERVO        'c'          // send triplet servo/function/position
#define RSP_SETSERVO        'd'          // acknowledge
#define CMD_SETTIMERS       'e'          // send new timers
#define RSP_SETTIMERS       'f'          // reply to send new timers
#define CMD_GETSERVOPOS     'g'          // request position of all servos for all functions
#define RSP_GETSERVOPOS     'h'          // reply position of all servos for all functions
#define CMD_MOVETOFUNC      'i'          // move servos to function
#define RSP_MOVETOFUNC      'j'          // acknowledge
#define CMD_GETSWVER        'k'          // request timer firmware version
#define RSP_GETSWVER        'l'          // reply to request timer firmware version
#define CMD_UNLOCK          'm'          // request enable motor
#define RSP_UNLOCK          'n'          // acknowledge enable motor
#define CMD_JUMP2           'o'          // request jump to function after early DT
#define RSP_JUMP2           'p'          // acknowledge
#define CMD_GETTMEMORY      'q'          // request current memory index
#define RSP_GETTMEMORY      'r'          // reply current memory index
#define CMD_SETTMEMORY      's'          // set memory index
#define RSP_SETTMEMORY      't'          // acknowledge
#define CMD_GETLOWRPM       'u'          // request idlethrottle
#define RSP_GETLOWRPM       'v'          // reply idlethrottle
#define CMD_SETLOWRPM       'w'          // send idlethrottle
#define RSP_SETLOWRPM       'x'          // acknowledge
```

General format

PREAMBLE (0x00)	SYNC (0x00 0x00 0x00 0x00)	LENGTH (1 byte)	COMMAND (1 byte)	DATA (variable length)	CHECKSUM (1 byte)	TERMINATOR (0x45)

Length: total number of bytes except the PREAMBLE, SYNC and TERMINATOR fields

Checksum: Sum MODULO 256 of all the bytes including LENGTH, COMMAND, DATA

CMD_GETSWVER – get software version

00 00 00 00 03 6B 6B 45

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789
SSSSL CT
YYYYE MHE
NNNNN  KR
00000664
00003BB5
```

RSP_GETSWVER – reply to get software version

(example B536 0A 04 03) where B=class (B,Q,E), 536=fw version, 0A=10 functions, 04 = 4 memories, 03 = 3 servos

00 00 00 00 0C 6C 42 35 33 36 20 0A 04 03 20 9D 45

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789
SSSSL           CT
YYYYE  9byte    HE
NNNNN           KR
000006          4
0000CC          5
```

CMD_GETTIMERS – request timers

00 00 00 00 03 61 61 45

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789
SSSSL CT
YYYYE MHE
NNNNN  KR
00000664
00003115
```

RSP_GETTIMERS – reply to request_timers

00 00 00 00 19 66 dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd CK 45

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789
SSSSL              CT
YYYYE  11x16bit timers    HE   !note: timers are 16bit values and express hundredths of seconds
NNNNN              KR
000016             4
000092             5
```

CMD_SETTIMERS – send new timers

`00 00 00 00 19 65 dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd  CK 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSL        11 16bit timers        CT
YYYYEM                              HE
NNNNN        11 16bit timers        KR
000016                              4
000095                              5
```

RSP_SETTIMERS – reply to send new_timers

`00 00 00 00 19 66 dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd dddd  CK 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSL        11 16bit timers        CT
YYYYEM                              HE
NNNNN                               KR
000016                              4
000096                              5
```

CMD_SETSERVO – send new servo position (function, servo, position)

`00 00 00 00 06 63 function servo position CHK 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLC    CT
YYYYEM    HE
NNNNN     KR
000006    4
000063    5
```

RSP_SETSERVO – reply to send new servo position

`00 00 00 00 03 64 64 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLCCT
YYYYEMHE
NNNNNKR
00000664
00003445
```

CMD_GETSERVOPOS – request servo positions

`00 00 00 00 03 67 67 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLCCT
YYYYEMHE
NNNNNKR
00000664
00003775
```

RSP_GETSERVOPOS – reply to request servo positions

`00 00 00 00 24 68 dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd CK 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLC                              CT   (NF+1 positions per servo, here NF = 10 and 3 servos -> total = 11 x 3 = 33 positions)
YYYYEM                              HE   !note: position varies from 0 to 130 (steps)
NNNNN                               KR
000026                              4
000048                              5
```

CMD_MOVETOFUNC – move servos to function

`00 00 00 00 04 69 function CHK 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLC CT
YYYYEM HE
NNNNN  KR
000006 4
000049 5
```

RSP_ MOVETOFUNC – reply to move servos to function

`00 00 00 00 03 6A 6A 45`

```
0         1         2         3         4         5         6         7
0123456789012345678901234567890123456789012345678901234567890123456789 0
SSSSLCCT
YYYYEMHE
NNNNNKR
00000664
00003AA5
```

CMD_UNLOCK – request enable motor
00 00 00 00 03 6D 6D 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000664
00003DD5
```

RSP_UNLOCK – reply to move servos to function
00 00 00 00 03 6A 6A 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000664
00003EE5
```

CMD_JUMP2 – request jump to function after early DT
00 00 00 00 03 6F 6F 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000664
00003FF5
```

RSP_JUMP2 – reply to jump to function after early DT
00 00 00 00 03 70 70 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003005
```

CMD_GETTMEMORY – request current memory index
00 00 00 00 03 71 71 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003115
```

RSP_GETTMEMORY – reply to request current memory index
00 00 00 00 03 72 72 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003225
```

CMD_SETTMEMORY – set current memory index
00 00 00 00 04 73 memory 73 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLC CT
YYYYPM HE
NNNNND KR
000007 74
000043 35
```

RSP_SETTMEMORY – reply to set current memory index
00 00 00 00 03 74 74 45

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003445
```

CMD_GETLOWRPM – request idle throttle

```
00 00 00 00 03 75 75 45
```

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003555
```

RSP_GETLOWRPM – reply to request idle throttle

```
00 00 00 00 04 76 rpm 76 45
```

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLC CT
YYYYPM HE
NNNNND KR
000007 74
000036 65
```

CMD_SETLOWRPM – request idle throttle

```
00 00 00 00 04 77 rpm 77 45
```

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLC CT
YYYYPM HE
NNNNND KR
000007 74
000047 75
```

RSP_SETLOWRPM – reply to request idle throttle

```
00 00 00 00 03 78 78 45
```

```
0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890
SSSSLCCT
YYYYPMHE
NNNNNDKR
00000774
00003885
```