# Blimp Instruction Set Architecture

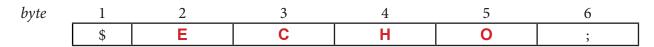
# **Instruction List**

ECHO
MOT—motor
PING
QRY—query
RST—reset
RUN
STAT—status
STOP

# **ECHO**

**DESCRIPTION:** reply to a PING message

COMMAND STRING: \$ECHO;



**LENGTH:** 6 bytes

**DIRECTION:** BLIMP  $\rightarrow$  PC

#### **NOTES:**

The ECHO command takes no arguments. This is a response in reply to a PING message from the PC.

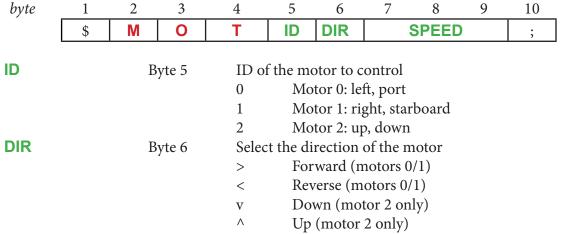
#### **EXAMPLE:**

\$ECHO;

# MOT—motor

**DESCRIPTION:** Low-level motor control commands

COMMAND STRING: \$MOT{ID}{DIR}{SPEED};



**SPEED** Bytes 7-9 Motor speed, as controlled by pwm duty cycle value

000 0% duty cycle; minimum speed

128 50% speed

255 100% duty cycle; maximum speed

LENGTH: 10 bytes

**DIRECTION:** PC → BLIMP

#### **NOTES:**

The low-level motor control commands control the speed of each of the three fans on the blimp.

It is an error to use v or ^ with a motor ID of 0 or 1. Likewise, > and < should only be used for motors 0 and 1 but not the up/down fan 2.

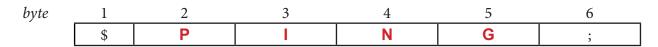
#### **EXAMPLE:**

```
$MOT1>128;  # right fan forward at 50% speed
$MOT2v000;  # vertical fan down at 0% speed (stopped)
$MOT0>255;  # left fan forward at full speed
$MOT2^016;  # vertical fan up at speed 16
$MOT0<064;  # left motor reverse at speed 64</pre>
```

# **PING**

**DESCRIPTION:** keepalive or ping message

COMMAND STRING: \$PING;



**LENGTH:** 6 bytes

**DIRECTION:** PC → BLIMP

#### **NOTES:**

The PING command takes no arguments. The blimp should send an ECHO command in response.

#### **EXAMPLE:**

\$PING;

# **QRY**—query

**DESCRIPTION:** Ask (query) for the status of a sub-system on the blimp

**COMMAND STRING**: \$QRY{SYSID};



**SYSID** Byte 5 ID of the system to query

S system status

M motion controller status

T temperature

**LENGTH:** 6 bytes

**DIRECTION:**  $PC \rightarrow BLIMP$ 

#### **NOTES:**

The QRY command asks the blimp about the status of a particular subsystem. The subsystems are usually abbreviated by a single upper-case letter, although any single byte can be used as long as both the PC and BLIMP recognize the command.

The blimp should respond to the QRY command with an appropriate STAT (status).

#### **EXAMPLE:**

\$QRYS; # query system status

\$QRYM; # query motion controller status

\$QRYT; # query temperature (implementation optional)

# RST—reset

**DESCRIPTION:** Disable sensors and motors

COMMAND STRING: \$RST;



LENGTH: 5 bytes

**DIRECTION:**  $PC \rightarrow BLIMP$ 

#### **NOTES:**

The RST command takes no arguments.

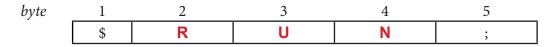
#### **EXAMPLE:**

\$RST;

# **RUN**

**DESCRIPTION:** Enable sensors and motors

COMMAND STRING: \$RUN;



LENGTH: 5 bytes

**DIRECTION:**  $PC \rightarrow BLIMP$ 

**NOTES:** 

The run command takes no arguments.

**EXAMPLE:** 

\$RUN;

### **STAT**—status

**DESCRIPTION:** Report the status of a subsystem

**COMMAND STRING**: \$STAT{SYSID}{STATE};



**SYSID** Byte 5 ID of the system to query

(see the QRY command for a list of possible SYSIDs)

**STATE** Bytes 6-7 State of the subsystem

UP up, subsystem running normal DN down, subsystem not running

ER error, subsystem error

LENGTH: 8 bytes

**DIRECTION:** BLIMP  $\rightarrow$  PC

#### **NOTES:**

The STAT command is usually in response to a QRY command, but it does not have to be. The blimp can send STAT commands to the PC at will if necessary.

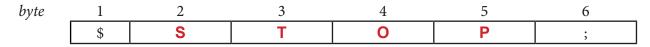
#### **EXAMPLE:**

\$STATSUP; # status of (main) system is up
\$STATMER; # motoion conroller has an error
\$STATTDN; # temperature subsystem is down

# **STOP**

**DESCRIPTION:** Disable sensors and motors

COMMAND STRING: \$STOP;



**LENGTH:** 6 bytes

**DIRECTION:**  $PC \rightarrow BLIMP$ 

#### **NOTES:**

The STOP command takes no arguments.

#### **EXAMPLE:**

\$STOP;