

StackMod Protocol

Paul Bupe Jr
Version 0.9.1

General Description

StackMod is a protocol created by Paul Bupe Jr to help facilitate the creation of modular robotics systems by abstracting the transfer of data between modules. This protocol is designed to be simple, human-readable, and is suitable for transmission over I2C or Serial. All data is passed as ASCII characters between 0x24 (36) and 0x7D (125).

Packet Structure

<START><ADDRESS><ACTION><COMMAND><DATA><END><CHECKSUM>

- The START byte is ASCII "{" (123 decimal, 0x7B).
- The ADDRESS byte is any value from 64 to 95 decimal (0x40 to 0x5F, ASCII "@" to "_").
- The ACTION byte is an ASCII "?" for GET and ASCII "\$" for SET.
- The COMMAND bytes are three characters.
- The DATA bytes are variable length and are described in the Data section.
- The END byte is ASCII "}" (125 decimal, 0x7D).
- The CHECKSUM is calculated by subtracting 32 from all the characters in the packet (excluding the checksum) and summing them. The modulo 95 of this value is then calculated and 32 is added back to that value.

```
int calculate_checksum(String packet) {  
    int sum = 0;  
    int c = packet.length();  
    for (int i = 0; i < c; i++) { sum += packet[i] - 32;}  
    return (sum % 95) + 32;  
}
```

Available Actions and Commands

The available commands for this revision are listed in the table below.

Command	Description
ARM	Arm / Disarm System
MTR	Motor Speed
SRV	Servo Position / Speed
ULT	Ultrasonic Distance
IRS	IR Value
DGT	Digital Pin
ANL	Analog Pin

MTR GET Action

The packet structure of the GET action for the MTR command is shown below. This packet requests the speed of motor number 2.

Command

{ Start
X Module address
? GET action
M Command byte 1
T Command byte 2
R Command byte 3
2 Motor number 2
} End
X Checksum value (88).

Response

{ Start
X Module address
? GET action
M Command byte 1
T Command byte 2
R Command byte 3
2 Motor number 2
+ Motor speed sign byte (+ or -)
0 Motor speed byte 1
7 Motor speed byte 2
8 Motor speed byte 3 (Allowed range is -100 to +100)
} End
C Checksum value (67).

MTR SET Action

Acceptable values are -100 to 100

Command

{ Start
X Module address
\$ SET action
M Command byte 1
T Command byte 2
R Command byte 3
2 Motor number 2
+ Motor speed sign byte (+ or -)
0 Motor speed byte 1
2 Motor speed byte 2
5 Motor speed byte 3 (Allowed range is -100 to +100)
} End
Checksum value (32 -- ASCII [space]).

SRV GET Action

The packet structure of the GET action for the SRV command is shown below. This packet requests the position of servo number 1.

Command

{	Start
X	Module address
?	GET action
S	Command byte 1
R	Command byte 2
V	Command byte 3
1	Servo number 1
}	End
–	Checksum value (95)

Response

{	Start
X	Module address
?	GET action
S	Command byte 1
R	Command byte 2
V	Command byte 3
1	Servo number 1
0	Servo position byte 1
3	Servo position byte 2
8	Servo position byte 3 (Allowed range is 0 to 180)
}	End
;	Checksum value (59)

SRV SET Action

Acceptable values are 0 to 180

Command

{	Start
X	Module address
\$	SET action
S	Command byte 1
R	Command byte 2
V	Command byte 3
1	Servo number 1
1	Servo position byte 1
6	Servo position byte 2
8	Servo position byte 3 (Allowed range is 0 to 180)
}	End
\$	Checksum value (36)