# BaseStation for Mongol

2.0

Generated by Doxygen 1.7.6.1

Sat Jun 16 2012 10:32:52

# Contents

# Chapter 1

# Data Structure Index

## 1.1 Data Structures

Here are the data structures with brief descriptions:

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 3

# Data Structure Documentation

## 3.1 packet_struct Struct Reference

`#include <BaseStation.h>`

### Data Fields

- byte **front_bnd**
- byte type
- byte value
- byte **end_bnd**
- byte null_term

### 3.1.1 Detailed Description

Packet structure definition.

Packet bounds should always be set to 0xFF. Packet type should be set to a PKT_Type. Values must be specified for every type except HELLO, GDBY, STDBY, and RDY.

### 3.1.2 Field Documentation

#### 3.1.2.1 byte packet_struct::null_term

Null terminator to ensure proper operation with string functions

#### 3.1.2.2 byte packet_struct::type

Type of packet

#### 3.1.2.3 byte packet_struct::value

Relevant value for type of packet

The documentation for this struct was generated from the following file:

- BaseStation.h

# Chapter 4

# File Documentation

## 4.1 BaseStation.h File Reference

```
#include <stdio.h>
```

### Data Structures

- struct packet_struct

### Typedefs

- typedef char **byte**
- typedef struct packet_struct packet

### Enumerations

- enum PKT_value { PKT_HELLO = 1, PKT_GDBY, PKT_STDBY, PKT_MOVE, PKT_TURN, PKT_AIM_H, PKT_AIM_V, PKT_FIRE, PKT_STRF_L, PKT_STRF_R, PKT_RDY }
- enum CTRL_mode { CTRL_KEYBOARD, CTRL_GAMEPAD }
- enum COMM_mode { COMM_ONLINE, COMM_OFFLINE }
- enum UI_mode { UI_CLINE, UI_GRAPH }
- enum PRINT_mode { PRNT_QUIET, PRNT_VERB }
- enum MOV_val { MOV_STOP = 1, MOV_FWD, MOV_BKD }
- enum TRN_val { TRN_NONE = 1, TRN_LEFT, TRN_RIGHT }
- enum AIM_H_val { AIM_H_STRGHT = 1, AIM_H_LEFT, AIM_H_RIGHT }
- enum AIM_V_val { AIM_V_STRGHT = 1, AIM_V_DWN, AIM_V_UP }
- enum FIRE_val { FIRE_ON = 1, FIRE_OFF }
- enum STRF_L_val { **STRF_L_OFF** = 1, **STRF_L_ON** }
- enum STRF_R_val { **STRF_R_OFF** = 1, **STRF_R_ON** }

### Functions

- void printmsg (void)
- void quit_base (void)

### Variables

- char termbuf [BUFSIZ]

## 4.1.1 Detailed Description

Main header file for BaseStation.

This header file contains the communication packet structure, enumerations for packet components, and enumerations for different modes of operation for the program.

## 4.1.2 Typedef Documentation

### 4.1.2.1 typedef struct packet_struct packet

Packet structure definition.

Packet bounds should always be set to 0xFF. Packet type should be set to a PKT_Type. Values must be specified for every type except HELLO, GDBY, STDBY, and RDY.

## 4.1.3 Enumeration Type Documentation

### 4.1.3.1 enum AIM_H_val

Horizontal robot turrent aim adjustment packet values.

This enumeration provides possible values for a packet of type PKT_AIM_H that will direct the robot to turn its turrent to either the left or right or not at all.

**Enumerator:**

***AIM_H_STRGHT*** Do not turn the turret in any direction

***AIM_H_LEFT*** Turn the turret to the left

***AIM_H_RIGHT*** Turn the turret to the right

### 4.1.3.2 enum AIM_V_val

Horizontal robot turrent aim adjustment packet values.

This enumeration provides possible values for a packet of type PKT_AIM_V that will direct the robot to raise, lower, or keep still its turret.

**Enumerator:**

***AIM_V_STRGHT*** Do not raise or lower the turret

***AIM_V_DWN*** Lower the turret

***AIM_V_UP*** Raise the turret

#### 4.1.3.3 enum COMM_mode

BaseStation-to-Robot communication mode.

This enumeration specifies possible modes for communicating with the robot.

**Enumerator:**

***COMM_ONLINE*** Attempt to establish connection to robot

***COMM_OFFLINE*** Do not attempt to establish connection to robot

#### 4.1.3.4 enum CTRL_mode

Controller mode values.

This enumeration specifies possible modes for controlling the robot.

**Enumerator:**

***CTRL_KEYBOARD*** Use keyboard keys to control robot

***CTRL_GAMEPAD*** Use gamepad's joystick and buttons to control robot

#### 4.1.3.5 enum FIRE_val

Robot turret firing mode values.

This enumeration provides possible values for a packet of type PKT_FIRE that will direct the robot to either fire or not fire the gun on its turret.

**Enumerator:**

***FIRE_ON*** Fire the gun

***FIRE_OFF*** Do not fire the gun

#### 4.1.3.6 enum MOV_val

Robot movement packet values.

This enumeration provides possible values for a packet of type PKT_MOVE that will direct the robot to move either forwards or backwards or not at all.

**Enumerator:**

***MOV_STOP*** Do not move the robot in any direction

***MOV_FWD*** Move the robot forward

***MOV_BKD*** Move the robot backward

#### 4.1.3.7 enum PKT_value

Available packet types.

These are the different types of packets available for communication with the robot. PKT_HELLO, PKT_GDBY, PKT_STDBY, and PKT_RDY are used to regulate communication and do not need corresponding values.

**Enumerator:**

***PKT_HELLO*** Hello packet: used to initialize communications

***PKT_GDBY*** Goodbye packet: used to close connections

***PKT_STDBY*** Standby packet: no robot updates (most common)

***PKT_MOVE*** Move packet: update moving

***PKT_TURN*** Turn packet: update turning

***PKT_AIM_H*** Horizontal aim packet: update horizontal aim adjustment

***PKT_AIM_V*** Vertical aim packet: update vertical aim adjustment

***PKT_FIRE*** Fire packet: update gun-firing status

***PKT_STRF_L*** Strafe packet: update left strafe status

***PKT_STRF_R*** Strafe packet: update right strafe status

***PKT_RDY*** Ready packet: indication that robot is ready for next packet

#### 4.1.3.8 enum PRINT_mode

Error and warning printing mode.

This enumeration specifies modes for quiet or verbose modes for reporting minor errors and warnings during runtime.

**Enumerator:**

***PRNT_QUIET*** Do not print messages about minor communication and input errors.

***PRNT_VERB*** Print messages about minor communication and input errors.

#### 4.1.3.9 enum STRF_L_val

Robot left-strafing packet values.

This enumeration provides possible values for a packet of type PKT_STRF_L that will direct the robot to strafe to the left or not.

#### 4.1.3.10 enum STRF_R_val

Robot right-strafing packet values.

This enumeration provides possible values for a packet of type PKT_STRF_L that will direct the robot to strafe to the right or not.

#### 4.1.3.11 enum TRN_val

Robot turning packet values.

This enumeration provides possible values for a packet of type PKT_TURN that will direct the robot to turn to the left or the right or not at all.

**Enumerator:**

***TRN_NONE*** Do not turn the robot in any direction

***TRN_LEFT*** Turn the robot to the left

***TRN_RIGHT*** Turn the robot to the right

#### 4.1.3.12 enum UI_mode

User interface mode.

This enumeration specifies modes for command-line interface or graphical interface.

**Enumerator:**

***UI_CLINE*** Do not initialize GUI, stay in command line.

***UI_GRAPH*** Initialize GUI.

### 4.1.4 Function Documentation

#### 4.1.4.1 void printmsg ( void )

Message printing/logging function.

This function will print and/or write to a log file error and warning messages, depending on what the print mode variable is set to.

#### 4.1.4.2 void quit_base ( void )

Quit BaseStation function.

This function properly closes all subsystems and communication channels before terminating the BaseStation program.

### 4.1.5 Variable Documentation

#### 4.1.5.1 char termbuf[BUFSIZ]

Message printing buffer