;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::BatteryDockingStatusStamped)

(if (not (find-package "ROBOTNIK\_MSGS\_NEW"))

(make-package "ROBOTNIK\_MSGS\_NEW"))

(shadow 'BatteryDockingStatusStamped (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::BATTERYDOCKINGSTATUSSTAMPED")

(make-package "ROBOTNIK\_MSGS\_NEW::BATTERYDOCKINGSTATUSSTAMPED"))

(in-package "ROS")

;;//! \htmlinclude BatteryDockingStatusStamped.msg.html

(if (not (find-package "STD\_MSGS"))

(ros::roseus-add-msgs "std\_msgs"))

(defclass robotnik\_msgs\_new::BatteryDockingStatusStamped

:super ros::object

:slots (\_header \_status ))

(defmethod robotnik\_msgs\_new::BatteryDockingStatusStamped

(:init

(&key

((:header \_\_header) (instance std\_msgs::Header :init))

((:status \_\_status) (instance robotnik\_msgs\_new::BatteryDockingStatus :init))

)

(send-super :init)

(setq \_header \_\_header)

(setq \_status \_\_status)

self)

(:header

(&rest \_\_header)

(if (keywordp (car \_\_header))

(send\* \_header \_\_header)

(progn

(if \_\_header (setq \_header (car \_\_header)))

\_header)))

(:status

(&rest \_\_status)

(if (keywordp (car \_\_status))

(send\* \_status \_\_status)

(progn

(if \_\_status (setq \_status (car \_\_status)))

\_status)))

(:serialization-length

()

(+

;; std\_msgs/Header \_header

(send \_header :serialization-length)

;; robotnik\_msgs\_new/BatteryDockingStatus \_status

(send \_status :serialization-length)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; std\_msgs/Header \_header

(send \_header :serialize s)

;; robotnik\_msgs\_new/BatteryDockingStatus \_status

(send \_status :serialize s)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; std\_msgs/Header \_header

(send \_header :deserialize buf ptr-) (incf ptr- (send \_header :serialization-length))

;; robotnik\_msgs\_new/BatteryDockingStatus \_status

(send \_status :deserialize buf ptr-) (incf ptr- (send \_status :serialization-length))

;;

self)

)

(setf (get robotnik\_msgs\_new::BatteryDockingStatusStamped :md5sum-) "f9b376e82e9d778484349573af188b1d")

(setf (get robotnik\_msgs\_new::BatteryDockingStatusStamped :datatype-) "robotnik\_msgs\_new/BatteryDockingStatusStamped")

(setf (get robotnik\_msgs\_new::BatteryDockingStatusStamped :definition-)

"Header header

BatteryDockingStatus status

================================================================================

MSG: std\_msgs/Header

# Standard metadata for higher-level stamped data types.

# This is generally used to communicate timestamped data

# in a particular coordinate frame.

#

# sequence ID: consecutively increasing ID

uint32 seq

#Two-integer timestamp that is expressed as:

# \* stamp.sec: seconds (stamp\_secs) since epoch (in Python the variable is called 'secs')

# \* stamp.nsec: nanoseconds since stamp\_secs (in Python the variable is called 'nsecs')

# time-handling sugar is provided by the client library

time stamp

#Frame this data is associated with

string frame\_id

================================================================================

MSG: robotnik\_msgs\_new/BatteryDockingStatus

# Modes of operation:

# no docking station contacts

string MODE\_DISABLED=disabled

# Unattended relay detection & activation with no inputs/outputs feedback. Done by the hw

string MODE\_AUTO\_HW=automatic\_hw

# Unattended relay detection & activation with inputs/outputs feedback. Done by the sw

string MODE\_AUTO\_SW=automatic\_sw

# Unattended relay detection & and manual activation of the charging relay

string MODE\_MANUAL\_SW=manual\_sw

string operation\_mode

bool contact\_relay\_status # shows if there's contact with the charger

bool charger\_relay\_status # shows if the relay for the charge is active or not

")

(provide :robotnik\_msgs\_new/BatteryDockingStatusStamped "f9b376e82e9d778484349573af188b1d")