;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::BatteryStatus

:super ros::object

:slots (\_voltage \_current \_level \_time\_remaining \_time\_charging \_is\_charging ))

(defmethod robotnik\_msgs\_new::BatteryStatus

(:init

(&key

((:voltage \_\_voltage) 0.0)

((:current \_\_current) 0.0)

((:level \_\_level) 0.0)

((:time\_remaining \_\_time\_remaining) 0)

((:time\_charging \_\_time\_charging) 0)

((:is\_charging \_\_is\_charging) nil)

)

(send-super :init)

(setq \_voltage (float \_\_voltage))

(setq \_current (float \_\_current))

(setq \_level (float \_\_level))

(setq \_time\_remaining (round \_\_time\_remaining))

(setq \_time\_charging (round \_\_time\_charging))

(setq \_is\_charging \_\_is\_charging)

self)

(:voltage

(&optional \_\_voltage)

(if \_\_voltage (setq \_voltage \_\_voltage)) \_voltage)

(:current

(&optional \_\_current)

(if \_\_current (setq \_current \_\_current)) \_current)

(:level

(&optional \_\_level)

(if \_\_level (setq \_level \_\_level)) \_level)

(:time\_remaining

(&optional \_\_time\_remaining)

(if \_\_time\_remaining (setq \_time\_remaining \_\_time\_remaining)) \_time\_remaining)

(:time\_charging

(&optional \_\_time\_charging)

(if \_\_time\_charging (setq \_time\_charging \_\_time\_charging)) \_time\_charging)

(:is\_charging

(&optional \_\_is\_charging)

(if \_\_is\_charging (setq \_is\_charging \_\_is\_charging)) \_is\_charging)

(:serialization-length

()

(+

;; float32 \_voltage

4

;; float32 \_current

4

;; float32 \_level

4

;; uint32 \_time\_remaining

4

;; uint32 \_time\_charging

4

;; bool \_is\_charging

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float32 \_voltage

(sys::poke \_voltage (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

;; float32 \_current

(sys::poke \_current (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

;; float32 \_level

(sys::poke \_level (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

;; uint32 \_time\_remaining

(write-long \_time\_remaining s)

;; uint32 \_time\_charging

(write-long \_time\_charging s)

;; bool \_is\_charging

(if \_is\_charging (write-byte -1 s) (write-byte 0 s))

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; float32 \_voltage

(setq \_voltage (sys::peek buf ptr- :float)) (incf ptr- 4)

;; float32 \_current

(setq \_current (sys::peek buf ptr- :float)) (incf ptr- 4)

;; float32 \_level

(setq \_level (sys::peek buf ptr- :float)) (incf ptr- 4)

;; uint32 \_time\_remaining

(setq \_time\_remaining (sys::peek buf ptr- :integer)) (incf ptr- 4)

;; uint32 \_time\_charging

(setq \_time\_charging (sys::peek buf ptr- :integer)) (incf ptr- 4)

;; bool \_is\_charging

(setq \_is\_charging (not (= 0 (sys::peek buf ptr- :char)))) (incf ptr- 1)

;;

self)

)

(setf (get robotnik\_msgs\_new::BatteryStatus :md5sum-) "6b1ba378a989551f0cb299f03c5355bb")

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

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

"float32 voltage # in volts

float32 current # in amperes

float32 level # in %

uint32 time\_remaining # in minutes

uint32 time\_charging # in minutes

bool is\_charging # true when connected

")

(provide :robotnik\_msgs\_new/BatteryStatus "6b1ba378a989551f0cb299f03c5355bb")