;; Auto-generated. Do not edit!

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

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

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

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

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

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

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

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

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

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

(in-package "ROS")

(defclass robotnik\_msgs\_new::SetMotorModeRequest

:super ros::object

:slots (\_mode ))

(defmethod robotnik\_msgs\_new::SetMotorModeRequest

(:init

(&key

((:mode \_\_mode) "")

)

(send-super :init)

(setq \_mode (string \_\_mode))

self)

(:mode

(&optional \_\_mode)

(if \_\_mode (setq \_mode \_\_mode)) \_mode)

(:serialization-length

()

(+

;; string \_mode

4 (length \_mode)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string \_mode

(write-long (length \_mode) s) (princ \_mode s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string \_mode

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_mode (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass robotnik\_msgs\_new::SetMotorModeResponse

:super ros::object

:slots (\_success \_message ))

(defmethod robotnik\_msgs\_new::SetMotorModeResponse

(:init

(&key

((:success \_\_success) nil)

((:message \_\_message) "")

)

(send-super :init)

(setq \_success \_\_success)

(setq \_message (string \_\_message))

self)

(:success

(&optional \_\_success)

(if \_\_success (setq \_success \_\_success)) \_success)

(:message

(&optional \_\_message)

(if \_\_message (setq \_message \_\_message)) \_message)

(:serialization-length

()

(+

;; bool \_success

1

;; string \_message

4 (length \_message)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_success

(if \_success (write-byte -1 s) (write-byte 0 s))

;; string \_message

(write-long (length \_message) s) (princ \_message s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_success

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

;; string \_message

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_message (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass robotnik\_msgs\_new::SetMotorMode

:super ros::object

:slots ())

(setf (get robotnik\_msgs\_new::SetMotorMode :md5sum-) "1393b8f659688f5f0d86f35948597ef1")

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

(setf (get robotnik\_msgs\_new::SetMotorMode :request) robotnik\_msgs\_new::SetMotorModeRequest)

(setf (get robotnik\_msgs\_new::SetMotorMode :response) robotnik\_msgs\_new::SetMotorModeResponse)

(defmethod robotnik\_msgs\_new::SetMotorModeRequest

(:response () (instance robotnik\_msgs\_new::SetMotorModeResponse :init)))

(setf (get robotnik\_msgs\_new::SetMotorModeRequest :md5sum-) "1393b8f659688f5f0d86f35948597ef1")

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

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

"string mode

---

bool success

string message

")

(setf (get robotnik\_msgs\_new::SetMotorModeResponse :md5sum-) "1393b8f659688f5f0d86f35948597ef1")

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

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

"string mode

---

bool success

string message

")

(provide :robotnik\_msgs\_new/SetMotorMode "1393b8f659688f5f0d86f35948597ef1")