;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(intern "\*STANDARD\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*STANDARD\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*STANDARD\* "standard")

(intern "\*DOCKING\_STATION\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*DOCKING\_STATION\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*DOCKING\_STATION\* "docking\_station")

(intern "\*CART\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*CART\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*CART\* "cart")

(intern "\*CART\_DOCKING\_CART\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*CART\_DOCKING\_CART\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*CART\_DOCKING\_CART\* "cart\_docking\_cart")

(intern "\*DOCKING\_CART\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*DOCKING\_CART\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*DOCKING\_CART\* "docking\_cart")

(intern "\*INVALID\*" (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(shadow '\*INVALID\* (find-package "ROBOTNIK\_MSGS\_NEW::LASERMODE"))

(defconstant robotnik\_msgs\_new::LaserMode::\*INVALID\* "invalid")

(defclass robotnik\_msgs\_new::LaserMode

:super ros::object

:slots (\_name ))

(defmethod robotnik\_msgs\_new::LaserMode

(:init

(&key

((:name \_\_name) "")

)

(send-super :init)

(setq \_name (string \_\_name))

self)

(:name

(&optional \_\_name)

(if \_\_name (setq \_name \_\_name)) \_name)

(:serialization-length

()

(+

;; string \_name

4 (length \_name)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string \_name

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string \_name

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

;;

self)

)

(setf (get robotnik\_msgs\_new::LaserMode :md5sum-) "6d9ec22e8c4a4279a8a61cc162391fee")

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

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

"string STANDARD=standard

string DOCKING\_STATION=docking\_station

string CART=cart

string CART\_DOCKING\_CART=cart\_docking\_cart

string DOCKING\_CART=docking\_cart

string INVALID=invalid

string name

")

(provide :robotnik\_msgs\_new/LaserMode "6d9ec22e8c4a4279a8a61cc162391fee")