;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::encoders

:super ros::object

:slots (\_type \_counts \_vel ))

(defmethod robotnik\_msgs\_new::encoders

(:init

(&key

((:type \_\_type) (let (r) (dotimes (i 0) (push "" r)) r))

((:counts \_\_counts) (make-array 0 :initial-element 0 :element-type :integer))

((:vel \_\_vel) (make-array 0 :initial-element 0 :element-type :integer))

)

(send-super :init)

(setq \_type \_\_type)

(setq \_counts \_\_counts)

(setq \_vel \_\_vel)

self)

(:type

(&optional \_\_type)

(if \_\_type (setq \_type \_\_type)) \_type)

(:counts

(&optional \_\_counts)

(if \_\_counts (setq \_counts \_\_counts)) \_counts)

(:vel

(&optional \_\_vel)

(if \_\_vel (setq \_vel \_\_vel)) \_vel)

(:serialization-length

()

(+

;; string[] \_type

(apply #'+ (mapcar #'(lambda (x) (+ 4 (length x))) \_type)) 4

;; int32[] \_counts

(\* 4 (length \_counts)) 4

;; int32[] \_vel

(\* 4 (length \_vel)) 4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string[] \_type

(write-long (length \_type) s)

(dolist (elem \_type)

(write-long (length elem) s) (princ elem s)

)

;; int32[] \_counts

(write-long (length \_counts) s)

(dotimes (i (length \_counts))

(write-long (elt \_counts i) s)

)

;; int32[] \_vel

(write-long (length \_vel) s)

(dotimes (i (length \_vel))

(write-long (elt \_vel i) s)

)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string[] \_type

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_type (make-list n))

(dotimes (i n)

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setf (elt \_type i) (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

))

;; int32[] \_counts

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_counts (instantiate integer-vector n))

(dotimes (i n)

(setf (elt \_counts i) (sys::peek buf ptr- :integer)) (incf ptr- 4)

))

;; int32[] \_vel

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_vel (instantiate integer-vector n))

(dotimes (i n)

(setf (elt \_vel i) (sys::peek buf ptr- :integer)) (incf ptr- 4)

))

;;

self)

)

(setf (get robotnik\_msgs\_new::encoders :md5sum-) "b5def35e667417de284b17ffe2083938")

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

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

"# 0-> right incremental, 1-> left incremental, 2-> steer absolute

string[] type

int32[] counts

int32[] vel

")

(provide :robotnik\_msgs\_new/encoders "b5def35e667417de284b17ffe2083938")