;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::MotorHeadingOffset

:super ros::object

:slots (\_motor \_value ))

(defmethod robotnik\_msgs\_new::MotorHeadingOffset

(:init

(&key

((:motor \_\_motor) 0)

((:value \_\_value) 0.0)

)

(send-super :init)

(setq \_motor (round \_\_motor))

(setq \_value (float \_\_value))

self)

(:motor

(&optional \_\_motor)

(if \_\_motor (setq \_motor \_\_motor)) \_motor)

(:value

(&optional \_\_value)

(if \_\_value (setq \_value \_\_value)) \_value)

(:serialization-length

()

(+

;; int32 \_motor

4

;; float64 \_value

8

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; int32 \_motor

(write-long \_motor s)

;; float64 \_value

(sys::poke \_value (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; int32 \_motor

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

;; float64 \_value

(setq \_value (sys::peek buf ptr- :double)) (incf ptr- 8)

;;

self)

)

(setf (get robotnik\_msgs\_new::MotorHeadingOffset :md5sum-) "8f9a9c9e1eb9b64236a3a4e805aa730d")

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

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

"int32 motor

float64 value

")

(provide :robotnik\_msgs\_new/MotorHeadingOffset "8f9a9c9e1eb9b64236a3a4e805aa730d")