;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::Axis

:super ros::object

:slots (\_pan \_tilt \_zoom \_focus \_brightness \_iris \_autofocus \_autoiris ))

(defmethod robotnik\_msgs\_new::Axis

(:init

(&key

((:pan \_\_pan) 0.0)

((:tilt \_\_tilt) 0.0)

((:zoom \_\_zoom) 0.0)

((:focus \_\_focus) 0.0)

((:brightness \_\_brightness) 0.0)

((:iris \_\_iris) 0.0)

((:autofocus \_\_autofocus) nil)

((:autoiris \_\_autoiris) nil)

)

(send-super :init)

(setq \_pan (float \_\_pan))

(setq \_tilt (float \_\_tilt))

(setq \_zoom (float \_\_zoom))

(setq \_focus (float \_\_focus))

(setq \_brightness (float \_\_brightness))

(setq \_iris (float \_\_iris))

(setq \_autofocus \_\_autofocus)

(setq \_autoiris \_\_autoiris)

self)

(:pan

(&optional \_\_pan)

(if \_\_pan (setq \_pan \_\_pan)) \_pan)

(:tilt

(&optional \_\_tilt)

(if \_\_tilt (setq \_tilt \_\_tilt)) \_tilt)

(:zoom

(&optional \_\_zoom)

(if \_\_zoom (setq \_zoom \_\_zoom)) \_zoom)

(:focus

(&optional \_\_focus)

(if \_\_focus (setq \_focus \_\_focus)) \_focus)

(:brightness

(&optional \_\_brightness)

(if \_\_brightness (setq \_brightness \_\_brightness)) \_brightness)

(:iris

(&optional \_\_iris)

(if \_\_iris (setq \_iris \_\_iris)) \_iris)

(:autofocus

(&optional \_\_autofocus)

(if \_\_autofocus (setq \_autofocus \_\_autofocus)) \_autofocus)

(:autoiris

(&optional \_\_autoiris)

(if \_\_autoiris (setq \_autoiris \_\_autoiris)) \_autoiris)

(:serialization-length

()

(+

;; float32 \_pan

4

;; float32 \_tilt

4

;; float32 \_zoom

4

;; float32 \_focus

4

;; float32 \_brightness

4

;; float32 \_iris

4

;; bool \_autofocus

1

;; bool \_autoiris

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float32 \_pan

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

;; float32 \_tilt

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

;; float32 \_zoom

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

;; float32 \_focus

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

;; float32 \_brightness

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

;; float32 \_iris

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

;; bool \_autofocus

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

;; bool \_autoiris

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; float32 \_pan

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

;; float32 \_tilt

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

;; float32 \_zoom

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

;; float32 \_focus

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

;; float32 \_brightness

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

;; float32 \_iris

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

;; bool \_autofocus

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

;; bool \_autoiris

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

;;

self)

)

(setf (get robotnik\_msgs\_new::Axis :md5sum-) "e5a99fe291a5635d3f85f17b629d8088")

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

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

"float32 pan

float32 tilt

float32 zoom

float32 focus

float32 brightness

float32 iris

bool autofocus

bool autoiris

")

(provide :robotnik\_msgs\_new/Axis "e5a99fe291a5635d3f85f17b629d8088")