;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::set\_analog\_output)

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

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

(shadow 'set\_analog\_output (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUT")

(make-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUT"))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUTREQUEST")

(make-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUTREQUEST"))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUTRESPONSE")

(make-package "ROBOTNIK\_MSGS\_NEW::SET\_ANALOG\_OUTPUTRESPONSE"))

(in-package "ROS")

(defclass robotnik\_msgs\_new::set\_analog\_outputRequest

:super ros::object

:slots (\_output \_value ))

(defmethod robotnik\_msgs\_new::set\_analog\_outputRequest

(:init

(&key

((:output \_\_output) 0)

((:value \_\_value) 0.0)

)

(send-super :init)

(setq \_output (round \_\_output))

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

self)

(:output

(&optional \_\_output)

(if \_\_output (setq \_output \_\_output)) \_output)

(:value

(&optional \_\_value)

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

(:serialization-length

()

(+

;; int8 \_output

1

;; float32 \_value

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; int8 \_output

(write-byte \_output s)

;; float32 \_value

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; int8 \_output

(setq \_output (sys::peek buf ptr- :char)) (incf ptr- 1)

(if (> \_output 127) (setq \_output (- \_output 256)))

;; float32 \_value

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

;;

self)

)

(defclass robotnik\_msgs\_new::set\_analog\_outputResponse

:super ros::object

:slots (\_ret ))

(defmethod robotnik\_msgs\_new::set\_analog\_outputResponse

(:init

(&key

((:ret \_\_ret) nil)

)

(send-super :init)

(setq \_ret \_\_ret)

self)

(:ret

(&optional \_\_ret)

(if \_\_ret (setq \_ret \_\_ret)) \_ret)

(:serialization-length

()

(+

;; bool \_ret

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_ret

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_ret

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

;;

self)

)

(defclass robotnik\_msgs\_new::set\_analog\_output

:super ros::object

:slots ())

(setf (get robotnik\_msgs\_new::set\_analog\_output :md5sum-) "1f857604e9e8f27790b5ef70d33fca08")

(setf (get robotnik\_msgs\_new::set\_analog\_output :datatype-) "robotnik\_msgs\_new/set\_analog\_output")

(setf (get robotnik\_msgs\_new::set\_analog\_output :request) robotnik\_msgs\_new::set\_analog\_outputRequest)

(setf (get robotnik\_msgs\_new::set\_analog\_output :response) robotnik\_msgs\_new::set\_analog\_outputResponse)

(defmethod robotnik\_msgs\_new::set\_analog\_outputRequest

(:response () (instance robotnik\_msgs\_new::set\_analog\_outputResponse :init)))

(setf (get robotnik\_msgs\_new::set\_analog\_outputRequest :md5sum-) "1f857604e9e8f27790b5ef70d33fca08")

(setf (get robotnik\_msgs\_new::set\_analog\_outputRequest :datatype-) "robotnik\_msgs\_new/set\_analog\_outputRequest")

(setf (get robotnik\_msgs\_new::set\_analog\_outputRequest :definition-)

"int8 output

float32 value

---

bool ret

")

(setf (get robotnik\_msgs\_new::set\_analog\_outputResponse :md5sum-) "1f857604e9e8f27790b5ef70d33fca08")

(setf (get robotnik\_msgs\_new::set\_analog\_outputResponse :datatype-) "robotnik\_msgs\_new/set\_analog\_outputResponse")

(setf (get robotnik\_msgs\_new::set\_analog\_outputResponse :definition-)

"int8 output

float32 value

---

bool ret

")

(provide :robotnik\_msgs\_new/set\_analog\_output "1f857604e9e8f27790b5ef70d33fca08")