;; Auto-generated. Do not edit!

(when (boundp 'ur\_msgs::Analog)

(if (not (find-package "UR\_MSGS"))

(make-package "UR\_MSGS"))

(shadow 'Analog (find-package "UR\_MSGS")))

(unless (find-package "UR\_MSGS::ANALOG")

(make-package "UR\_MSGS::ANALOG"))

(in-package "ROS")

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

(defclass ur\_msgs::Analog

:super ros::object

:slots (\_pin \_state ))

(defmethod ur\_msgs::Analog

(:init

(&key

((:pin \_\_pin) 0)

((:state \_\_state) 0.0)

)

(send-super :init)

(setq \_pin (round \_\_pin))

(setq \_state (float \_\_state))

self)

(:pin

(&optional \_\_pin)

(if \_\_pin (setq \_pin \_\_pin)) \_pin)

(:state

(&optional \_\_state)

(if \_\_state (setq \_state \_\_state)) \_state)

(:serialization-length

()

(+

;; uint8 \_pin

1

;; float32 \_state

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; uint8 \_pin

(write-byte \_pin s)

;; float32 \_state

(sys::poke \_state (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))

;; uint8 \_pin

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

;; float32 \_state

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

;;

self)

)

(setf (get ur\_msgs::Analog :md5sum-) "341541c8828d055b6dcc443d40207a7d")

(setf (get ur\_msgs::Analog :datatype-) "ur\_msgs/Analog")

(setf (get ur\_msgs::Analog :definition-)

"uint8 pin

float32 state

")

(provide :ur\_msgs/Analog "341541c8828d055b6dcc443d40207a7d")