;; Auto-generated. Do not edit!

(when (boundp 'schunk\_ezn64::PosAndSpeedFloat32)

(if (not (find-package "SCHUNK\_EZN64"))

(make-package "SCHUNK\_EZN64"))

(shadow 'PosAndSpeedFloat32 (find-package "SCHUNK\_EZN64")))

(unless (find-package "SCHUNK\_EZN64::POSANDSPEEDFLOAT32")

(make-package "SCHUNK\_EZN64::POSANDSPEEDFLOAT32"))

(in-package "ROS")

;;//! \htmlinclude PosAndSpeedFloat32.msg.html

(defclass schunk\_ezn64::PosAndSpeedFloat32

:super ros::object

:slots (\_pos \_speed ))

(defmethod schunk\_ezn64::PosAndSpeedFloat32

(:init

(&key

((:pos \_\_pos) 0.0)

((:speed \_\_speed) 0.0)

)

(send-super :init)

(setq \_pos (float \_\_pos))

(setq \_speed (float \_\_speed))

self)

(:pos

(&optional \_\_pos)

(if \_\_pos (setq \_pos \_\_pos)) \_pos)

(:speed

(&optional \_\_speed)

(if \_\_speed (setq \_speed \_\_speed)) \_speed)

(:serialization-length

()

(+

;; float32 \_pos

4

;; float32 \_speed

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float32 \_pos

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

;; float32 \_speed

(sys::poke \_speed (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))

;; float32 \_pos

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

;; float32 \_speed

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

;;

self)

)

(setf (get schunk\_ezn64::PosAndSpeedFloat32 :md5sum-) "0611d9adb487c3bcf130e5b7b3ee7f37")

(setf (get schunk\_ezn64::PosAndSpeedFloat32 :datatype-) "schunk\_ezn64/PosAndSpeedFloat32")

(setf (get schunk\_ezn64::PosAndSpeedFloat32 :definition-)

"float32 pos

float32 speed

")

(provide :schunk\_ezn64/PosAndSpeedFloat32 "0611d9adb487c3bcf130e5b7b3ee7f37")