;; Auto-generated. Do not edit!

(when (boundp 'schunk\_ezn64::set\_velocity)

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

(make-package "SCHUNK\_EZN64"))

(shadow 'set\_velocity (find-package "SCHUNK\_EZN64")))

(unless (find-package "SCHUNK\_EZN64::SET\_VELOCITY")

(make-package "SCHUNK\_EZN64::SET\_VELOCITY"))

(unless (find-package "SCHUNK\_EZN64::SET\_VELOCITYREQUEST")

(make-package "SCHUNK\_EZN64::SET\_VELOCITYREQUEST"))

(unless (find-package "SCHUNK\_EZN64::SET\_VELOCITYRESPONSE")

(make-package "SCHUNK\_EZN64::SET\_VELOCITYRESPONSE"))

(in-package "ROS")

(defclass schunk\_ezn64::set\_velocityRequest

:super ros::object

:slots (\_goal\_velocity ))

(defmethod schunk\_ezn64::set\_velocityRequest

(:init

(&key

((:goal\_velocity \_\_goal\_velocity) 0.0)

)

(send-super :init)

(setq \_goal\_velocity (float \_\_goal\_velocity))

self)

(:goal\_velocity

(&optional \_\_goal\_velocity)

(if \_\_goal\_velocity (setq \_goal\_velocity \_\_goal\_velocity)) \_goal\_velocity)

(:serialization-length

()

(+

;; float32 \_goal\_velocity

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float32 \_goal\_velocity

(sys::poke \_goal\_velocity (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 \_goal\_velocity

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

;;

self)

)

(defclass schunk\_ezn64::set\_velocityResponse

:super ros::object

:slots (\_goal\_accepted ))

(defmethod schunk\_ezn64::set\_velocityResponse

(:init

(&key

((:goal\_accepted \_\_goal\_accepted) nil)

)

(send-super :init)

(setq \_goal\_accepted \_\_goal\_accepted)

self)

(:goal\_accepted

(&optional (\_\_goal\_accepted :null))

(if (not (eq \_\_goal\_accepted :null)) (setq \_goal\_accepted \_\_goal\_accepted)) \_goal\_accepted)

(:serialization-length

()

(+

;; bool \_goal\_accepted

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_goal\_accepted

(if \_goal\_accepted (write-byte -1 s) (write-byte 0 s))

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_goal\_accepted

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

;;

self)

)

(defclass schunk\_ezn64::set\_velocity

:super ros::object

:slots ())

(setf (get schunk\_ezn64::set\_velocity :md5sum-) "8bcbe89faecc3b1502a1b95a8574cab4")

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

(setf (get schunk\_ezn64::set\_velocity :request) schunk\_ezn64::set\_velocityRequest)

(setf (get schunk\_ezn64::set\_velocity :response) schunk\_ezn64::set\_velocityResponse)

(defmethod schunk\_ezn64::set\_velocityRequest

(:response () (instance schunk\_ezn64::set\_velocityResponse :init)))

(setf (get schunk\_ezn64::set\_velocityRequest :md5sum-) "8bcbe89faecc3b1502a1b95a8574cab4")

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

(setf (get schunk\_ezn64::set\_velocityRequest :definition-)

"float32 goal\_velocity

---

bool goal\_accepted

")

(setf (get schunk\_ezn64::set\_velocityResponse :md5sum-) "8bcbe89faecc3b1502a1b95a8574cab4")

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

(setf (get schunk\_ezn64::set\_velocityResponse :definition-)

"float32 goal\_velocity

---

bool goal\_accepted

")

(provide :schunk\_ezn64/set\_velocity "8bcbe89faecc3b1502a1b95a8574cab4")