;; Auto-generated. Do not edit!

(when (boundp 'schunk\_ezn64::acknowledge\_error)

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

(make-package "SCHUNK\_EZN64"))

(shadow 'acknowledge\_error (find-package "SCHUNK\_EZN64")))

(unless (find-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERROR")

(make-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERROR"))

(unless (find-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERRORREQUEST")

(make-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERRORREQUEST"))

(unless (find-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERRORRESPONSE")

(make-package "SCHUNK\_EZN64::ACKNOWLEDGE\_ERRORRESPONSE"))

(in-package "ROS")

(defclass schunk\_ezn64::acknowledge\_errorRequest

:super ros::object

:slots ())

(defmethod schunk\_ezn64::acknowledge\_errorRequest

(:init

(&key

)

(send-super :init)

self)

(:serialization-length

()

(+

0

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;;

self)

)

(defclass schunk\_ezn64::acknowledge\_errorResponse

:super ros::object

:slots (\_acknowledge\_response ))

(defmethod schunk\_ezn64::acknowledge\_errorResponse

(:init

(&key

((:acknowledge\_response \_\_acknowledge\_response) nil)

)

(send-super :init)

(setq \_acknowledge\_response \_\_acknowledge\_response)

self)

(:acknowledge\_response

(&optional (\_\_acknowledge\_response :null))

(if (not (eq \_\_acknowledge\_response :null)) (setq \_acknowledge\_response \_\_acknowledge\_response)) \_acknowledge\_response)

(:serialization-length

()

(+

;; bool \_acknowledge\_response

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_acknowledge\_response

(if \_acknowledge\_response (write-byte -1 s) (write-byte 0 s))

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_acknowledge\_response

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

;;

self)

)

(defclass schunk\_ezn64::acknowledge\_error

:super ros::object

:slots ())

(setf (get schunk\_ezn64::acknowledge\_error :md5sum-) "216312424a7b13297dba83de8033144e")

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

(setf (get schunk\_ezn64::acknowledge\_error :request) schunk\_ezn64::acknowledge\_errorRequest)

(setf (get schunk\_ezn64::acknowledge\_error :response) schunk\_ezn64::acknowledge\_errorResponse)

(defmethod schunk\_ezn64::acknowledge\_errorRequest

(:response () (instance schunk\_ezn64::acknowledge\_errorResponse :init)))

(setf (get schunk\_ezn64::acknowledge\_errorRequest :md5sum-) "216312424a7b13297dba83de8033144e")

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

(setf (get schunk\_ezn64::acknowledge\_errorRequest :definition-)

"---

bool acknowledge\_response

")

(setf (get schunk\_ezn64::acknowledge\_errorResponse :md5sum-) "216312424a7b13297dba83de8033144e")

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

(setf (get schunk\_ezn64::acknowledge\_errorResponse :definition-)

"---

bool acknowledge\_response

")

(provide :schunk\_ezn64/acknowledge\_error "216312424a7b13297dba83de8033144e")