;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::Cartesian\_Euler\_pose)

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

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

(shadow 'Cartesian\_Euler\_pose (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::CARTESIAN\_EULER\_POSE")

(make-package "ROBOTNIK\_MSGS\_NEW::CARTESIAN\_EULER\_POSE"))

(in-package "ROS")

;;//! \htmlinclude Cartesian\_Euler\_pose.msg.html

(defclass robotnik\_msgs\_new::Cartesian\_Euler\_pose

:super ros::object

:slots (\_x \_y \_z \_A \_B \_C ))

(defmethod robotnik\_msgs\_new::Cartesian\_Euler\_pose

(:init

(&key

((:x \_\_x) 0.0)

((:y \_\_y) 0.0)

((:z \_\_z) 0.0)

((:A \_\_A) 0.0)

((:B \_\_B) 0.0)

((:C \_\_C) 0.0)

)

(send-super :init)

(setq \_x (float \_\_x))

(setq \_y (float \_\_y))

(setq \_z (float \_\_z))

(setq \_A (float \_\_A))

(setq \_B (float \_\_B))

(setq \_C (float \_\_C))

self)

(:x

(&optional \_\_x)

(if \_\_x (setq \_x \_\_x)) \_x)

(:y

(&optional \_\_y)

(if \_\_y (setq \_y \_\_y)) \_y)

(:z

(&optional \_\_z)

(if \_\_z (setq \_z \_\_z)) \_z)

(:A

(&optional \_\_A)

(if \_\_A (setq \_A \_\_A)) \_A)

(:B

(&optional \_\_B)

(if \_\_B (setq \_B \_\_B)) \_B)

(:C

(&optional \_\_C)

(if \_\_C (setq \_C \_\_C)) \_C)

(:serialization-length

()

(+

;; float64 \_x

8

;; float64 \_y

8

;; float64 \_z

8

;; float64 \_A

8

;; float64 \_B

8

;; float64 \_C

8

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float64 \_x

(sys::poke \_x (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_y

(sys::poke \_y (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_z

(sys::poke \_z (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_A

(sys::poke \_A (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_B

(sys::poke \_B (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_C

(sys::poke \_C (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; float64 \_x

(setq \_x (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_y

(setq \_y (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_z

(setq \_z (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_A

(setq \_A (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_B

(setq \_B (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_C

(setq \_C (sys::peek buf ptr- :double)) (incf ptr- 8)

;;

self)

)

(setf (get robotnik\_msgs\_new::Cartesian\_Euler\_pose :md5sum-) "1b7d5ac5679ead09b31ec87e784aa10e")

(setf (get robotnik\_msgs\_new::Cartesian\_Euler\_pose :datatype-) "robotnik\_msgs\_new/Cartesian\_Euler\_pose")

(setf (get robotnik\_msgs\_new::Cartesian\_Euler\_pose :definition-)

"float64 x

float64 y

float64 z

float64 A

float64 B

float64 C

")

(provide :robotnik\_msgs\_new/Cartesian\_Euler\_pose "1b7d5ac5679ead09b31ec87e784aa10e")