;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::inputs\_outputs)

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

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

(shadow 'inputs\_outputs (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::INPUTS\_OUTPUTS")

(make-package "ROBOTNIK\_MSGS\_NEW::INPUTS\_OUTPUTS"))

(in-package "ROS")

;;//! \htmlinclude inputs\_outputs.msg.html

(defclass robotnik\_msgs\_new::inputs\_outputs

:super ros::object

:slots (\_digital\_inputs \_digital\_outputs \_analog\_inputs \_analog\_outputs ))

(defmethod robotnik\_msgs\_new::inputs\_outputs

(:init

(&key

((:digital\_inputs \_\_digital\_inputs) (let (r) (dotimes (i 0) (push nil r)) r))

((:digital\_outputs \_\_digital\_outputs) (let (r) (dotimes (i 0) (push nil r)) r))

((:analog\_inputs \_\_analog\_inputs) (make-array 0 :initial-element 0.0 :element-type :float))

((:analog\_outputs \_\_analog\_outputs) (make-array 0 :initial-element 0.0 :element-type :float))

)

(send-super :init)

(setq \_digital\_inputs \_\_digital\_inputs)

(setq \_digital\_outputs \_\_digital\_outputs)

(setq \_analog\_inputs \_\_analog\_inputs)

(setq \_analog\_outputs \_\_analog\_outputs)

self)

(:digital\_inputs

(&optional \_\_digital\_inputs)

(if \_\_digital\_inputs (setq \_digital\_inputs \_\_digital\_inputs)) \_digital\_inputs)

(:digital\_outputs

(&optional \_\_digital\_outputs)

(if \_\_digital\_outputs (setq \_digital\_outputs \_\_digital\_outputs)) \_digital\_outputs)

(:analog\_inputs

(&optional \_\_analog\_inputs)

(if \_\_analog\_inputs (setq \_analog\_inputs \_\_analog\_inputs)) \_analog\_inputs)

(:analog\_outputs

(&optional \_\_analog\_outputs)

(if \_\_analog\_outputs (setq \_analog\_outputs \_\_analog\_outputs)) \_analog\_outputs)

(:serialization-length

()

(+

;; bool[] \_digital\_inputs

(\* 1 (length \_digital\_inputs)) 4

;; bool[] \_digital\_outputs

(\* 1 (length \_digital\_outputs)) 4

;; float32[] \_analog\_inputs

(\* 4 (length \_analog\_inputs)) 4

;; float32[] \_analog\_outputs

(\* 4 (length \_analog\_outputs)) 4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool[] \_digital\_inputs

(write-long (length \_digital\_inputs) s)

(dotimes (i (length \_digital\_inputs))

(if (elt \_digital\_inputs i) (write-byte -1 s) (write-byte 0 s))

)

;; bool[] \_digital\_outputs

(write-long (length \_digital\_outputs) s)

(dotimes (i (length \_digital\_outputs))

(if (elt \_digital\_outputs i) (write-byte -1 s) (write-byte 0 s))

)

;; float32[] \_analog\_inputs

(write-long (length \_analog\_inputs) s)

(dotimes (i (length \_analog\_inputs))

(sys::poke (elt \_analog\_inputs i) (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

)

;; float32[] \_analog\_outputs

(write-long (length \_analog\_outputs) s)

(dotimes (i (length \_analog\_outputs))

(sys::poke (elt \_analog\_outputs i) (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))

;; bool[] \_digital\_inputs

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_digital\_inputs (make-list n))

(dotimes (i n)

(setf (elt \_digital\_inputs i) (not (= 0 (sys::peek buf ptr- :char)))) (incf ptr- 1)

))

;; bool[] \_digital\_outputs

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_digital\_outputs (make-list n))

(dotimes (i n)

(setf (elt \_digital\_outputs i) (not (= 0 (sys::peek buf ptr- :char)))) (incf ptr- 1)

))

;; float32[] \_analog\_inputs

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_analog\_inputs (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_analog\_inputs i) (sys::peek buf ptr- :float)) (incf ptr- 4)

))

;; float32[] \_analog\_outputs

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_analog\_outputs (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_analog\_outputs i) (sys::peek buf ptr- :float)) (incf ptr- 4)

))

;;

self)

)

(setf (get robotnik\_msgs\_new::inputs\_outputs :md5sum-) "219be75bd56b8ebdca0dfa526c3803be")

(setf (get robotnik\_msgs\_new::inputs\_outputs :datatype-) "robotnik\_msgs\_new/inputs\_outputs")

(setf (get robotnik\_msgs\_new::inputs\_outputs :definition-)

"bool[] digital\_inputs

bool[] digital\_outputs

float32[] analog\_inputs

float32[] analog\_outputs

")

(provide :robotnik\_msgs\_new/inputs\_outputs "219be75bd56b8ebdca0dfa526c3803be")