;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::SetMotorPID)

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

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

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

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

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

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

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

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

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

(in-package "ROS")

(defclass robotnik\_msgs\_new::SetMotorPIDRequest

:super ros::object

:slots (\_pid ))

(defmethod robotnik\_msgs\_new::SetMotorPIDRequest

(:init

(&key

((:pid \_\_pid) (instance robotnik\_msgs\_new::MotorPID :init))

)

(send-super :init)

(setq \_pid \_\_pid)

self)

(:pid

(&rest \_\_pid)

(if (keywordp (car \_\_pid))

(send\* \_pid \_\_pid)

(progn

(if \_\_pid (setq \_pid (car \_\_pid)))

\_pid)))

(:serialization-length

()

(+

;; robotnik\_msgs\_new/MotorPID \_pid

(send \_pid :serialization-length)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; robotnik\_msgs\_new/MotorPID \_pid

(send \_pid :serialize s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; robotnik\_msgs\_new/MotorPID \_pid

(send \_pid :deserialize buf ptr-) (incf ptr- (send \_pid :serialization-length))

;;

self)

)

(defclass robotnik\_msgs\_new::SetMotorPIDResponse

:super ros::object

:slots (\_success \_message ))

(defmethod robotnik\_msgs\_new::SetMotorPIDResponse

(:init

(&key

((:success \_\_success) nil)

((:message \_\_message) "")

)

(send-super :init)

(setq \_success \_\_success)

(setq \_message (string \_\_message))

self)

(:success

(&optional \_\_success)

(if \_\_success (setq \_success \_\_success)) \_success)

(:message

(&optional \_\_message)

(if \_\_message (setq \_message \_\_message)) \_message)

(:serialization-length

()

(+

;; bool \_success

1

;; string \_message

4 (length \_message)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_success

(if \_success (write-byte -1 s) (write-byte 0 s))

;; string \_message

(write-long (length \_message) s) (princ \_message s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_success

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

;; string \_message

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_message (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass robotnik\_msgs\_new::SetMotorPID

:super ros::object

:slots ())

(setf (get robotnik\_msgs\_new::SetMotorPID :md5sum-) "b471f23ed4110be72a67eb0fa77abae1")

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

(setf (get robotnik\_msgs\_new::SetMotorPID :request) robotnik\_msgs\_new::SetMotorPIDRequest)

(setf (get robotnik\_msgs\_new::SetMotorPID :response) robotnik\_msgs\_new::SetMotorPIDResponse)

(defmethod robotnik\_msgs\_new::SetMotorPIDRequest

(:response () (instance robotnik\_msgs\_new::SetMotorPIDResponse :init)))

(setf (get robotnik\_msgs\_new::SetMotorPIDRequest :md5sum-) "b471f23ed4110be72a67eb0fa77abae1")

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

(setf (get robotnik\_msgs\_new::SetMotorPIDRequest :definition-)

"MotorPID pid

================================================================================

MSG: robotnik\_msgs\_new/MotorPID

# either can\_id or name are set

# if can\_id is -1, then this refers to all motors.

int32[] can\_id

string[] name

float32[] kp

float32[] ki

float32[] kd

---

bool success

string message

")

(setf (get robotnik\_msgs\_new::SetMotorPIDResponse :md5sum-) "b471f23ed4110be72a67eb0fa77abae1")

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

(setf (get robotnik\_msgs\_new::SetMotorPIDResponse :definition-)

"MotorPID pid

================================================================================

MSG: robotnik\_msgs\_new/MotorPID

# either can\_id or name are set

# if can\_id is -1, then this refers to all motors.

int32[] can\_id

string[] name

float32[] kp

float32[] ki

float32[] kd

---

bool success

string message

")

(provide :robotnik\_msgs\_new/SetMotorPID "b471f23ed4110be72a67eb0fa77abae1")