;; Auto-generated. Do not edit!

(when (boundp 'ur\_dashboard\_msgs::SetModeGoal)

(if (not (find-package "UR\_DASHBOARD\_MSGS"))

(make-package "UR\_DASHBOARD\_MSGS"))

(shadow 'SetModeGoal (find-package "UR\_DASHBOARD\_MSGS")))

(unless (find-package "UR\_DASHBOARD\_MSGS::SETMODEGOAL")

(make-package "UR\_DASHBOARD\_MSGS::SETMODEGOAL"))

(in-package "ROS")

;;//! \htmlinclude SetModeGoal.msg.html

(defclass ur\_dashboard\_msgs::SetModeGoal

:super ros::object

:slots (\_target\_robot\_mode \_stop\_program \_play\_program ))

(defmethod ur\_dashboard\_msgs::SetModeGoal

(:init

(&key

((:target\_robot\_mode \_\_target\_robot\_mode) (instance ur\_dashboard\_msgs::RobotMode :init))

((:stop\_program \_\_stop\_program) nil)

((:play\_program \_\_play\_program) nil)

)

(send-super :init)

(setq \_target\_robot\_mode \_\_target\_robot\_mode)

(setq \_stop\_program \_\_stop\_program)

(setq \_play\_program \_\_play\_program)

self)

(:target\_robot\_mode

(&rest \_\_target\_robot\_mode)

(if (keywordp (car \_\_target\_robot\_mode))

(send\* \_target\_robot\_mode \_\_target\_robot\_mode)

(progn

(if \_\_target\_robot\_mode (setq \_target\_robot\_mode (car \_\_target\_robot\_mode)))

\_target\_robot\_mode)))

(:stop\_program

(&optional \_\_stop\_program)

(if \_\_stop\_program (setq \_stop\_program \_\_stop\_program)) \_stop\_program)

(:play\_program

(&optional \_\_play\_program)

(if \_\_play\_program (setq \_play\_program \_\_play\_program)) \_play\_program)

(:serialization-length

()

(+

;; ur\_dashboard\_msgs/RobotMode \_target\_robot\_mode

(send \_target\_robot\_mode :serialization-length)

;; bool \_stop\_program

1

;; bool \_play\_program

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; ur\_dashboard\_msgs/RobotMode \_target\_robot\_mode

(send \_target\_robot\_mode :serialize s)

;; bool \_stop\_program

(if \_stop\_program (write-byte -1 s) (write-byte 0 s))

;; bool \_play\_program

(if \_play\_program (write-byte -1 s) (write-byte 0 s))

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; ur\_dashboard\_msgs/RobotMode \_target\_robot\_mode

(send \_target\_robot\_mode :deserialize buf ptr-) (incf ptr- (send \_target\_robot\_mode :serialization-length))

;; bool \_stop\_program

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

;; bool \_play\_program

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

;;

self)

)

(setf (get ur\_dashboard\_msgs::SetModeGoal :md5sum-) "a32b63f0e8a660b2b96916617d699c39")

(setf (get ur\_dashboard\_msgs::SetModeGoal :datatype-) "ur\_dashboard\_msgs/SetModeGoal")

(setf (get ur\_dashboard\_msgs::SetModeGoal :definition-)

"# ====== DO NOT MODIFY! AUTOGENERATED FROM AN ACTION DEFINITION ======

# This action is for setting the robot into a desired mode (e.g. RUNNING) and safety mode into a

# non-critical state (e.g. NORMAL or REDUCED), for example after a safety incident happened.

# goal

ur\_dashboard\_msgs/RobotMode target\_robot\_mode

# Stop program execution before restoring the target mode. Can be used together with 'play\_program'.

bool stop\_program

# Play the currently loaded program after target mode is reached.#

# NOTE: Requesting mode RUNNING in combination with this will make the robot continue the motion it

# was doing before. This might probably lead into the same problem (protective stop, EM-Stop due to

# faulty motion, etc.) If you want to be safe, set the 'stop\_program' flag below and manually play

# the program after robot state is returned to normal.

# This flag will only be used when requesting mode RUNNING

bool play\_program

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

MSG: ur\_dashboard\_msgs/RobotMode

int8 NO\_CONTROLLER=-1

int8 DISCONNECTED=0

int8 CONFIRM\_SAFETY=1

int8 BOOTING=2

int8 POWER\_OFF=3

int8 POWER\_ON=4

int8 IDLE=5

int8 BACKDRIVE=6

int8 RUNNING=7

int8 UPDATING\_FIRMWARE=8

int8 mode

")

(provide :ur\_dashboard\_msgs/SetModeGoal "a32b63f0e8a660b2b96916617d699c39")