;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass ur\_dashboard\_msgs::SetModeAction

:super ros::object

:slots (\_action\_goal \_action\_result \_action\_feedback ))

(defmethod ur\_dashboard\_msgs::SetModeAction

(:init

(&key

((:action\_goal \_\_action\_goal) (instance ur\_dashboard\_msgs::SetModeActionGoal :init))

((:action\_result \_\_action\_result) (instance ur\_dashboard\_msgs::SetModeActionResult :init))

((:action\_feedback \_\_action\_feedback) (instance ur\_dashboard\_msgs::SetModeActionFeedback :init))

)

(send-super :init)

(setq \_action\_goal \_\_action\_goal)

(setq \_action\_result \_\_action\_result)

(setq \_action\_feedback \_\_action\_feedback)

self)

(:action\_goal

(&rest \_\_action\_goal)

(if (keywordp (car \_\_action\_goal))

(send\* \_action\_goal \_\_action\_goal)

(progn

(if \_\_action\_goal (setq \_action\_goal (car \_\_action\_goal)))

\_action\_goal)))

(:action\_result

(&rest \_\_action\_result)

(if (keywordp (car \_\_action\_result))

(send\* \_action\_result \_\_action\_result)

(progn

(if \_\_action\_result (setq \_action\_result (car \_\_action\_result)))

\_action\_result)))

(:action\_feedback

(&rest \_\_action\_feedback)

(if (keywordp (car \_\_action\_feedback))

(send\* \_action\_feedback \_\_action\_feedback)

(progn

(if \_\_action\_feedback (setq \_action\_feedback (car \_\_action\_feedback)))

\_action\_feedback)))

(:serialization-length

()

(+

;; ur\_dashboard\_msgs/SetModeActionGoal \_action\_goal

(send \_action\_goal :serialization-length)

;; ur\_dashboard\_msgs/SetModeActionResult \_action\_result

(send \_action\_result :serialization-length)

;; ur\_dashboard\_msgs/SetModeActionFeedback \_action\_feedback

(send \_action\_feedback :serialization-length)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; ur\_dashboard\_msgs/SetModeActionGoal \_action\_goal

(send \_action\_goal :serialize s)

;; ur\_dashboard\_msgs/SetModeActionResult \_action\_result

(send \_action\_result :serialize s)

;; ur\_dashboard\_msgs/SetModeActionFeedback \_action\_feedback

(send \_action\_feedback :serialize s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; ur\_dashboard\_msgs/SetModeActionGoal \_action\_goal

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

;; ur\_dashboard\_msgs/SetModeActionResult \_action\_result

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

;; ur\_dashboard\_msgs/SetModeActionFeedback \_action\_feedback

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

;;

self)

)

(setf (get ur\_dashboard\_msgs::SetModeAction :md5sum-) "7048f11efd68ac8c7a2750e444f501df")

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

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

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

SetModeActionGoal action\_goal

SetModeActionResult action\_result

SetModeActionFeedback action\_feedback

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

MSG: ur\_dashboard\_msgs/SetModeActionGoal

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

Header header

actionlib\_msgs/GoalID goal\_id

SetModeGoal goal

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

MSG: std\_msgs/Header

# Standard metadata for higher-level stamped data types.

# This is generally used to communicate timestamped data

# in a particular coordinate frame.

#

# sequence ID: consecutively increasing ID

uint32 seq

#Two-integer timestamp that is expressed as:

# \* stamp.sec: seconds (stamp\_secs) since epoch (in Python the variable is called 'secs')

# \* stamp.nsec: nanoseconds since stamp\_secs (in Python the variable is called 'nsecs')

# time-handling sugar is provided by the client library

time stamp

#Frame this data is associated with

string frame\_id

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

MSG: actionlib\_msgs/GoalID

# The stamp should store the time at which this goal was requested.

# It is used by an action server when it tries to preempt all

# goals that were requested before a certain time

time stamp

# The id provides a way to associate feedback and

# result message with specific goal requests. The id

# specified must be unique.

string id

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

MSG: ur\_dashboard\_msgs/SetModeGoal

# ====== 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

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

MSG: ur\_dashboard\_msgs/SetModeActionResult

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

Header header

actionlib\_msgs/GoalStatus status

SetModeResult result

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

MSG: actionlib\_msgs/GoalStatus

GoalID goal\_id

uint8 status

uint8 PENDING = 0 # The goal has yet to be processed by the action server

uint8 ACTIVE = 1 # The goal is currently being processed by the action server

uint8 PREEMPTED = 2 # The goal received a cancel request after it started executing

# and has since completed its execution (Terminal State)

uint8 SUCCEEDED = 3 # The goal was achieved successfully by the action server (Terminal State)

uint8 ABORTED = 4 # The goal was aborted during execution by the action server due

# to some failure (Terminal State)

uint8 REJECTED = 5 # The goal was rejected by the action server without being processed,

# because the goal was unattainable or invalid (Terminal State)

uint8 PREEMPTING = 6 # The goal received a cancel request after it started executing

# and has not yet completed execution

uint8 RECALLING = 7 # The goal received a cancel request before it started executing,

# but the action server has not yet confirmed that the goal is canceled

uint8 RECALLED = 8 # The goal received a cancel request before it started executing

# and was successfully cancelled (Terminal State)

uint8 LOST = 9 # An action client can determine that a goal is LOST. This should not be

# sent over the wire by an action server

#Allow for the user to associate a string with GoalStatus for debugging

string text

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

MSG: ur\_dashboard\_msgs/SetModeResult

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

# result

bool success

string message

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

MSG: ur\_dashboard\_msgs/SetModeActionFeedback

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

Header header

actionlib\_msgs/GoalStatus status

SetModeFeedback feedback

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

MSG: ur\_dashboard\_msgs/SetModeFeedback

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

# feedback

int8 current\_robot\_mode

int8 current\_safety\_mode

")

(provide :ur\_dashboard\_msgs/SetModeAction "7048f11efd68ac8c7a2750e444f501df")