;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::Register

:super ros::object

:slots (\_id \_value ))

(defmethod robotnik\_msgs\_new::Register

(:init

(&key

((:id \_\_id) 0)

((:value \_\_value) 0)

)

(send-super :init)

(setq \_id (round \_\_id))

(setq \_value (round \_\_value))

self)

(:id

(&optional \_\_id)

(if \_\_id (setq \_id \_\_id)) \_id)

(:value

(&optional \_\_value)

(if \_\_value (setq \_value \_\_value)) \_value)

(:serialization-length

()

(+

;; int32 \_id

4

;; int32 \_value

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; int32 \_id

(write-long \_id s)

;; int32 \_value

(write-long \_value s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; int32 \_id

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

;; int32 \_value

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

;;

self)

)

(setf (get robotnik\_msgs\_new::Register :md5sum-) "b971379644cd29fdf0000430f7d4b8e8")

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

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

"int32 id

int32 value

")

(provide :robotnik\_msgs\_new/Register "b971379644cd29fdf0000430f7d4b8e8")