;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::get\_modbus\_register)

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

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

(shadow 'get\_modbus\_register (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTER")

(make-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTER"))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTERREQUEST")

(make-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTERREQUEST"))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTERRESPONSE")

(make-package "ROBOTNIK\_MSGS\_NEW::GET\_MODBUS\_REGISTERRESPONSE"))

(in-package "ROS")

(defclass robotnik\_msgs\_new::get\_modbus\_registerRequest

:super ros::object

:slots (\_address ))

(defmethod robotnik\_msgs\_new::get\_modbus\_registerRequest

(:init

(&key

((:address \_\_address) 0)

)

(send-super :init)

(setq \_address (round \_\_address))

self)

(:address

(&optional \_\_address)

(if \_\_address (setq \_address \_\_address)) \_address)

(:serialization-length

()

(+

;; int32 \_address

4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; int32 \_address

(write-long \_address s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; int32 \_address

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

;;

self)

)

(defclass robotnik\_msgs\_new::get\_modbus\_registerResponse

:super ros::object

:slots (\_ret \_value ))

(defmethod robotnik\_msgs\_new::get\_modbus\_registerResponse

(:init

(&key

((:ret \_\_ret) nil)

((:value \_\_value) 0)

)

(send-super :init)

(setq \_ret \_\_ret)

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

self)

(:ret

(&optional \_\_ret)

(if \_\_ret (setq \_ret \_\_ret)) \_ret)

(:value

(&optional \_\_value)

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

(:serialization-length

()

(+

;; bool \_ret

1

;; uint16 \_value

2

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; bool \_ret

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

;; uint16 \_value

(write-word \_value s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; bool \_ret

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

;; uint16 \_value

(setq \_value (sys::peek buf ptr- :short)) (incf ptr- 2)

;;

self)

)

(defclass robotnik\_msgs\_new::get\_modbus\_register

:super ros::object

:slots ())

(setf (get robotnik\_msgs\_new::get\_modbus\_register :md5sum-) "3a2192c58e10e29a7d1d9e8d896473ab")

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

(setf (get robotnik\_msgs\_new::get\_modbus\_register :request) robotnik\_msgs\_new::get\_modbus\_registerRequest)

(setf (get robotnik\_msgs\_new::get\_modbus\_register :response) robotnik\_msgs\_new::get\_modbus\_registerResponse)

(defmethod robotnik\_msgs\_new::get\_modbus\_registerRequest

(:response () (instance robotnik\_msgs\_new::get\_modbus\_registerResponse :init)))

(setf (get robotnik\_msgs\_new::get\_modbus\_registerRequest :md5sum-) "3a2192c58e10e29a7d1d9e8d896473ab")

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

(setf (get robotnik\_msgs\_new::get\_modbus\_registerRequest :definition-)

"int32 address

---

bool ret

uint16 value

")

(setf (get robotnik\_msgs\_new::get\_modbus\_registerResponse :md5sum-) "3a2192c58e10e29a7d1d9e8d896473ab")

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

(setf (get robotnik\_msgs\_new::get\_modbus\_registerResponse :definition-)

"int32 address

---

bool ret

uint16 value

")

(provide :robotnik\_msgs\_new/get\_modbus\_register "3a2192c58e10e29a7d1d9e8d896473ab")