Case 3:06-cv-01905-JSW Document 96 Filed 09/11/2006 Page 1 of 27

Exhibit B



(12) United States Patent Katzer

(10) Patent No.:

US 6,530,329 B2

(45) Date of Patent:

*Mar. 11, 2003

(54)	MODEL	TRAIN	CONTROL.	CVCTEM

Inventor: Matthew A. Katzer, 1416 NW. Benfield Dr., Portland, OR (US) 97229

Assignee: Matthew A. Katzer, Hillsboro, OR

(US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35

U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: 10/124,878

(22)Filed: Apr. 17, 2002

(65)**Prior Publication Data**

US 2002/0170458 A1 Nov. 21, 2002

Related U.S. Application Data

(63)	Continuation of application No. 09/858,222, filed on A	pr.
` '	17, 2002, now Pat. No. 6,460,467.	•

(51)	Int. Cl. ⁷	A63	H	19/00
------	-----------------------	-----	---	-------

U.S. Cl. 105/1.5; 246/167 R; 246/197;

.... 105/1.5, 1.4, 29.2; Field of Search 246/187 A, 167 R, 197, 62, 701/20

(56)References Cited

U.S. PATENT DOCUMENTS

3,944,986 A	3/1976	Staples	
3,976,272 A	8/1976	Murray et al.	
4,307,302 A	12/1981	Russell	
4,853,883 A	* 8/1989	Nickles et al	348/121
5 072 000 A	12/1001	Molon	

5,475,818	Α		12/1995	Molyneaux et al 701/20
5,493,642	Α		2/1996	Dunsmuir et al.
5,638,522	Α		6/1997	Dunsmuir et al.
5,681,015	Α	*	10/1997	Kull 246/167 R
5,696,689	Α		12/1997	Okumura et al.
5,787,371	Α	*	7/1998	Balukin et al 246/187 A
5,828,979	Α		10/1998	Ploivka et al.
5,896,017	Α		4/1999	Severson et al.
5,940,005	Α		8/1999	Severson et al.
5,952,797	Α		9/1999	Rossier
6,065,406	Α	*	5/2000	Katzer 105/1.4
6,267,061	B 1		7/2001	Katzer
6,270,040	B 1		8/2001	Katzer

OTHER PUBLICATIONS

Chapell, David, Understanding ActiveX and OLE, 1996, Microsoft Press, Redmond.

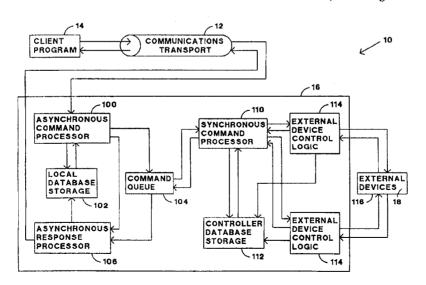
* cited by examiner

Primary Examiner-William A. Cuchlinski, Jr. Assistant Examiner—Olga Hernandez (74) Attorney, Agent, or Firm-Chernoff, Vilhauer, McClung & Stenzel, LLP

ABSTRACT

A system which operates a digitally controlled model railroad transmitting a first command from a first client program to a resident external controlling interface through a first communications transport. A second command is transmitted from a second client program to the resident external controlling interface through a second communications transport. The first command and the second command are received by the resident external controlling interface which queues the first and second commands. The resident external controlling interface sends third and fourth commands representative of the first and second commands, respectively, to a digital command station for execution on the digitally controlled model railroad.

27 Claims, 3 Drawing Sheets

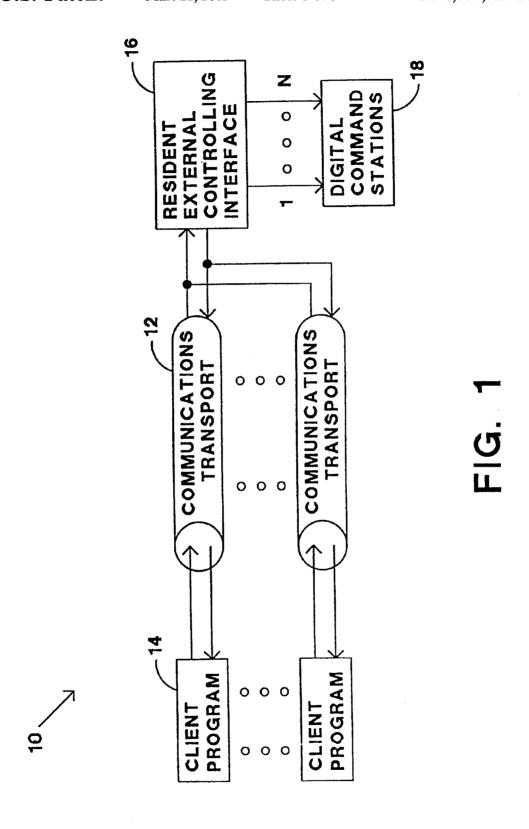


U.S. Patent

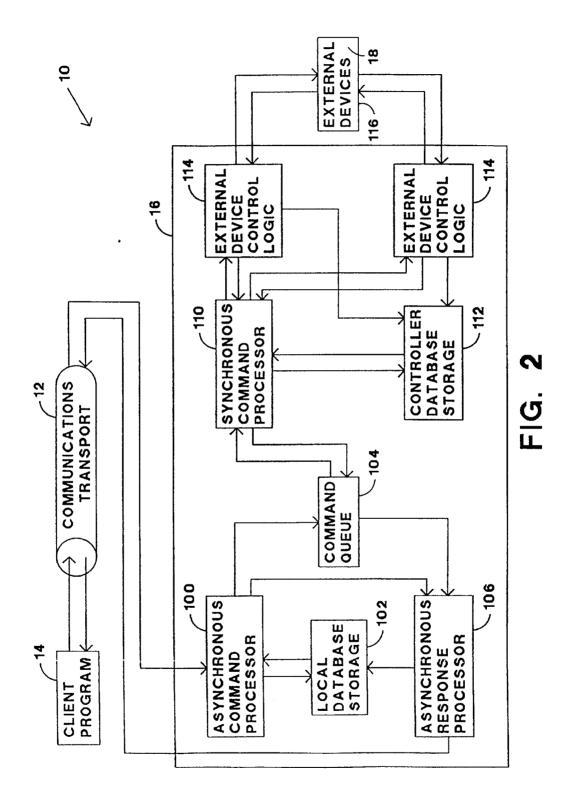
Mar. 11, 2003

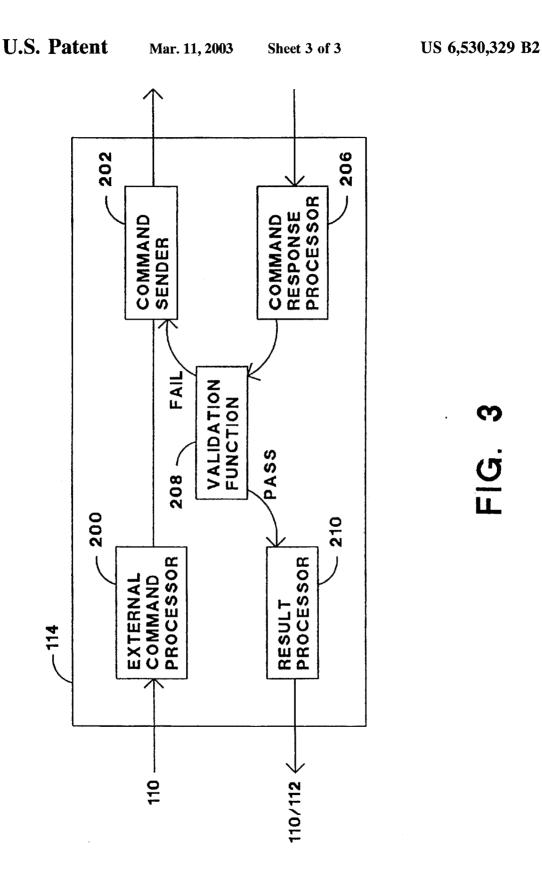
Sheet 1 of 3

US 6,530,329 B2



U.S. Patent Mar. 11, 2003 Sheet 2 of 3





1

MODEL TRAIN CONTROL SYSTEM

This application is a continuation of U.S. patent application Ser. No. 09/858,222 filed on Apr. 17, 2002 U.S. Pat. No. 6.460.467.

BACKGROUND OF THE INVENTION

The present invention relates to a system for controlling a model railroad.

Model railroads have traditionally been constructed with of a set of interconnected sections of train track, electric switches between different sections of the train track, and other electrically operated devices, such as train engines and draw bridges. Train engines receive their power to travel on the train track by electricity provided by a controller through the track itself. The speed and direction of the train engine is controlled by the level and polarity, respectively, of the electrical power supplied to the train track. The operator manually pushes buttons or pulls levers to cause the 20 switches or other electrically operated devices to function, as desired. Such model railroad sets are suitable for a single operator, but unfortunately they lack the capability of adequately controlling multiple trains independently. In addition, such model railroad sets are not suitable for being 25 controlled by multiple operators, especially if the operators are located at different locations distant from the model railroad, such as different cities.

A digital command control (DDC) system has been developed to provide additional controllability of individual train 30 engines and other electrical devices. Each device the operator desires to control, such as a train engine, includes an individually addressable digital decoder. A digital command station (DCS) is electrically connected to the train track to provide a command in the form of a set of encoded digital bits to a particular device that includes a digital decoder. The digital command station is typically controlled by a personal computer. A suitable standard for the digital command control system is the NMRA DCC Standards, issued March 1997, and is incorporated herein by reference. While pro- 40 viding the ability to individually control different devices of the railroad set, the DCC system still fails to provide the capability for multiple operators to control the railroad devices, especially if the operators are remotely located from the railroad set and each other.

DigiToys Systems of Lawrenceville, Ga. has developed a software program for controlling a model railroad set from a remote location. The software includes an interface which allows the operator to select desired changes to devices of the railroad set that include a digital decoder, such as increasing the speed of a train or switching a switch. The software issues a command locally or through a network, such as the internet, to a digital command station at the railroad set which executes the command. The protocol used by the software is based on Cobra from Open Management 55 Group where the software issues a command to a communication interface and awaits confirmation that the command was executed by the digital command station. When the software receives confirmation that the command executed, the software program sends the next command through the 60 communication interface to the digital command station. In other words, the technique used by the software to control the model railroad is analogous to an inexpensive printer where commands are sequentially issued to the printer after the previous command has been executed. Unfortunately, it has been observed that the response of the model railroad to the operator appears slow, especially over a distributed

2

network such as the internet. One technique to decrease the response time is to use high-speed network connections but unfortunately such connections are expensive.

What is desired, therefore, is a system for controlling a model railroad that effectively provides a high-speed connection without the additional expense associated therewith.

The foregoing and other objectives, features, and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

SUMMARY OF THE PRESENT INVENTION

The present invention overcomes the aforementioned drawbacks of the prior art, in a first aspect, by providing a system for operating a digitally controlled model railroad, that includes transmitting a first command from a first client program to a resident external controlling interface through a first communications transport. A second command is transmitted from a second client program to the resident external controlling interface through a second communications transport. The first command and the second command are received by the resident external controlling interface which queues the first and second commands. The resident external controlling interface sends third and fourth commands representative of the first and second commands, respectively, to a digital command station for execution on the digitally controller model railroad.

Incorporating a communications transport between the multiple client program and the resident external controlling interface permits multiple operators of the model railroad at locations distant from the physical model railroad and each other. In the environment of a model railroad club where the members want to simultaneously control devices of the same model railroad layout, which preferably includes multiple trains operating thereon, the operators each provide commands to the resistant external controlling interface, and hence the model railroad In addition by queuing by commands at a single resident external controlling interface permits controlled execution of the commands by the digitally controlled model railroad, would may otherwise conflict with one another.

In another aspect of the present invention the first command is selectively processed and sent to one of a plurality
of digital command stations for execution on the digitally
controlled model railroad based upon information contained
therein. Preferably, the second command is also selectively
processed and sent to one of the plurality of digital command
stations for execution on the digitally controlled model
railroad based upon information contained therein. The
resident external controlling interface also preferably
includes a command queue to maintain the order of the
commands.

The command queue also allows the sharing of multiple devices, multiple clients to communicate with the same device (locally or remote) in a controlled manner, and multiple clients to communicate with different devices. In other words, the command queue permits the proper execution in the cases of: (1) one client to many devices, (2) many clients to one device, and (3) many clients to many devices.

In yet another aspect of the present invention the first command is transmitted from a first client program to a first processor through a first communications transport. The first command is received at the first processor. The first processor provides an acknowledgement to the first client program through the first communications transport indicating that

3

the first command has properly executed prior to execution of commands related to the first command by the digitally controlled model railroad. The communications transport is preferably a COM or DCOM interface.

The model railroad application involves the use of 5 extremely slow real-time interfaces between the digital command stations and the devices of the model railroad. In order to increase the apparent speed of execution to the client, other than using high-speed communication interfaces, the resident external controller interface receives 10 the command and provides an acknowledgement to the client program in a timely manner before the execution of the command by the digital command stations. Accordingly, the execution of commands provided by the resident external controlling interface to the digital command stations 15 occur in a synchronous manner, such as a first-in-first-out manner. The COM and DCOM communications transport between the client program and the resident external controlling interface is operated in an asynchronous manner, namely providing an acknowledgement thereby releasing $^{\,20}$ the communications transport to accept further communications prior to the actual execution of the command. The combination of the synchronous and the asynchronous data communication for the commands provides the benefit that the operator considers the commands to occur nearly instan- 25 taneously while permitting the resident external controlling interface to verify that the command is proper and cause the commands to execute in a controlled manner by the digital command stations, all without additional high-speed communication networks. Moreover, for traditional distributed 30 software execution there is no motivation to provide an acknowledgment prior to the execution of the command because the command executes quickly and most commands are sequential in nature. In other words, the execution of the next command is dependent upon proper execution of the 35 prior command so there would be no motivation to provide an acknowledgment prior to its actual execution.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

FIG. 1 is a block diagram of an exemplary embodiment of a model train control system.

FIG. 2 is a more detailed block diagram of the model train control system of FIG. 1 including external device control $_{45}$ logic.

FIG. 3 is a block diagram of the external device control logic of FIG. 2.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 1, a model train control system 10 includes a communications transport 12 interconnecting a client program 14 and a resident external controlling interface 16. The client program 14 executes on the model 55 railroad operator's computer and may include any suitable system to permit the operator to provide desired commands to the resident external controlling interface 16. For example, the client program 14 may include a graphical interface representative of the model railroad layout where 60 the operator issues commands to the model railroad by making changes to the graphical interface. The client program 14 also defines a set of Application Programming Interfaces (API's), described in detail later, which the operator accesses using the graphical interface or other programs 65 such as Visual Basic, C++, Java, or browser based applications. There may be multiple client programs interconnected

4

with the resident external controlling interface 16 so that multiple remote operators may simultaneously provide control commands to the model railroad.

The communications transport 12 provides an interface between the client program 14 and the resident external controlling interface 16. The communications transport 12 may be any suitable communications medium for the transmission of data, such as the internet, local area network, satellite links, or multiple processes operating on a single computer. The preferred interface to the communications transport 12 is a COM or DCOM interface, as developed for the Windows operating system available from Microsoft Corporation. The communications transport 12 also determines if the resident external controlling interface 16 is system resident or remotely located on an external system. The communications transport 12 may also use private or public communications protocol as a medium for communications. The client program 14 provides commands and the resident external controlling interface 16 responds to the communications transport 12 to exchange information. A description of COM (common object model) and DCOM (distributed common object model) is provided by Chappel in a book entitled Understanding ActiveX and OLE, Microsoft Press, and is incorporated by reference herein.

Incorporating a communications transport 12 between the client program(s) 14 and the resident external controlling interface 16 permits multiple operators of the model railroad at locations distant from the physical model railroad and each other. In the environment of a model railroad club where the members want to simultaneously control devices of the same model railroad layout, which preferably includes multiple trains operating thereon, the operators each provide commands to the resistant external controlling interface, and hence the model railroad.

The manner in which commands are executed for the model railroad under COM and DCOM may be as follows. The client program 14 makes requests in a synchronous manner using COM/DCOM to the resident external interface controller 16. The synchronous manner of the request is the technique used by COM and DCOM to execute commands. The communications transport 12 packages the command for the transport mechanism to the resident external controlling interface 16. The resident external controlling interface 16 then passes the command to the digital command stations 18 which in turn executes the command. After the digital command station 18 executes the command an acknowledgement is passed back to the resident external controlling interface 16 which in turn passes an acknowledgement to the client program 14. Upon receipt of the 50 acknowledgement by the client program 14, the communications transport 12 is again available to accept another command. The train control system 10, without more, permits execution of commands by the digital command stations 18 from multiple operators, but like the DigiToys Systems' software the execution of commands is slow.

The present inventor came to the realization that unlike traditional distributed systems where the commands passed through a communications transport are executed nearly instantaneously by the server and then an acknowledgement is returned to the client, the model railroad application involves the use of extremely slow real-time interfaces between the digital command stations and the devices of the model railroad. The present inventor came to the further realization that in order to increase the apparent speed of execution to the client, other than using high-speed communication interfaces, the resident external controller interface 16 should receive the command and provide an

acknowledgement to the client program 12 in a timely manner before the execution of the command by the digital command stations 18. Accordingly, the execution of commands provided by the resident external controlling interface 16 to the digital command stations 18 occur in a synchronous manner, such as a first-in-first-out manner. The COM and DCOM communications transport 12 between the client program 14 and the resident external controlling interface 16 is operated in an asynchronous manner, namely providing an acknowledgement thereby releasing the communications transport 12 to accept further communications prior to the actual execution of the command. The combination of the synchronous and the asynchronous data communication for the commands provides the benefit that the operator considers the commands to occur nearly instantaneously while permitting the resident external controlling interface 16 to verify that the command is proper and cause the commands to execute in a controlled manner by the digital command stations 18, all without additional highspeed communication networks. Moreover, for traditional 20 distributed software execution there is no motivation to provide an acknowledgment prior to the execution of the command because the command executes quickly and most commands are sequential in nature. In other words, the execution of the next command is dependent upon proper execution of the prior command so there would be no motivation to provide an acknowledgment prior to its actual execution. It is to be understood that other devices, such as digital devices, may be controlled in a manner as described for model railroads.

Referring to FIG. 2, the client program 14 sends a command over the communications transport 12 that is received by an asynchronous command processor 100. The asynchronous command processor 100 queries a local database storage 102 to determine if it is necessary to package 35 a command to be transmitted to a command queue 104. The local database storage 102 primarily contains the state of the devices of the model railroad, such as for example, the speed of a train, the direction of a train, whether a draw bridge is up or down, whether a light is turned on or off, and the 40 configuration of the model railroad layout. If the command received by the asynchronous command processor 100 is a query of the state of a device, then the asynchronous command processor 100 retrieves such information from the local database storage 102 and provides the information to 45 an asynchronous response processor 106. The asynchronous response processor 106 then provides a response to the client program 14 indicating the state of the device and releases the communications transport 12 for the next command

The asynchronous command processor 100 also verifies, 50 using the configuration information in the local database storage 102, that the command received is a potentially valid operation. If the command is invalid, the asynchronous command processor 100 provides such information to the asynchronous response processor 106, which in turn returns 55 an error indication to the client program 14

The asynchronous command processor 100 may determine that the necessary information is not contained in the local database storage 102 to provide a response to the client program 14 of the device state or that the command is a valid 60 action. Actions may include, for example, an increase in the train's speed, or turning on/off of a device. In either case, the valid unknown state or action command is packaged and forwarded to the command queue 104. The packaging of the command may also include additional information from the 65 local database storage 102 to complete the client program 14 request, if necessary. Together with packaging the command

for the command queue 104, the asynchronous command processor 100 provides a command to the asynchronous request processor 106 to provide a response to the client program 14 indicating that the event has occurred, even though such an event has yet to occur on the physical railroad layout.

As such, it can be observed that whether or not the command is valid, whether or not the information requested by the command is available to the asynchronous command processor 100, and whether or not the command has executed, the combination of the asynchronous command processor 100 and the asynchronous response processor 106 both verifies the validity of the command and provides a response to the client program 14 thereby freeing up the communications transport 12 for additional commands. Without the asynchronous nature of the resident external controlling interface 16, the response to the client program 14 would be, in many circumstances, delayed thereby resulting in frustration to the operator that the model railroad is performing in a slow and painstaking manner. In this manner, the railroad operation using the asynchronous interface appears to the operator as nearly instantaneously responsive.

Each command in the command queue 104 is fetched by a synchronous command processor 110 and processed. The synchronous command processor 110 queries a controller database storage 112 for additional information, as necessary, and determines if the command has already been executed based on the state of the devices in the controller database storage 112. In the event that the command has already been executed, as indicated by the controller database storage 112, then the synchronous command processor 110 passes information to the command queue 104 that the command has been executed or the state of the device. The asynchronous response processor 106 fetches the information from the command cue 104 and provides a suitable response to the client program 14, if necessary, and updates the local database storage 102 to reflect the updated status of the railroad layout devices.

If the command fetched by the synchronous command processor 110 from the command queue 104 requires execution by external devices, such as the train engine, then the command is posted to one of several external device control logic 114 blocks. The external device control logic 114 processes the command from the synchronous command processor 110 and issues appropriate control commands to the interface of the particular external device 116 to execute the command on the device and ensure that an appropriate response was received in response. The external device is preferably a digital command control device that transmits digital commands to decoders using the train track. There are several different manufacturers of digital command stations, each of which has a different set of input commands, so each external device is designed for a particular digital command station. In this manner, the system is compatible with different digital command stations. The digital command stations 18 of the external devices 116 provide a response to the external device control logic 114 which is checked for validity and identified as to which prior command it corresponds to so that the controller database storage 112 may be updated properly. The process of transmitting commands to and receiving responses from the external devices 116 is slow.

The synchronous command processor 110 is notified of the results from the external control logic 114 and, if appropriate, forwards the results to the command queue 104. The asynchronous response processor 100 clears the results

from the command queue 104 and updates the local database storage 102 and sends an asynchronous response to the client program 14, if needed. The response updates the client program 14 of the actual state of the railroad track devices, if changed, and provides an error message to the client program 14 if the devices actual state was previously improperly reported or a command did not execute properly.

The use of two separate database storages, each of which is substantially a mirror image of the other, provides a performance enhancement by a fast acknowledgement to the 10 client program 14 using the local database storage 102 and thereby freeing up the communications transport 12 for additional commands. In addition, the number of commands forwarded to the external device control logic 114 and the external devices 116, which are relatively slow to respond, 15 is minimized by maintaining information concerning the state and configuration of the model railroad. Also, the use of two separate database tables 102 and 112 allows more efficient multi-threading on multi-processor computers.

In order to achieve the separation of the asynchronous and 20 synchronous portions of the system the command queue 104 is implemented as a named pipe, as developed by Microsoft for Windows. The queue 104 allows both portions to be separate from each other, where each considers the other to be the destination device. In addition, the command queue maintains the order of operation which is important to proper operation of the system.

The use of a single command queue 104 allows multiple instantrations of the asynchronous functionality, with one for each different client. The single command queue 104 also allows the sharing of multiple devices, multiple clients to communicate with the same device (locally or remote) in a controlled manner, and multiple clients to communicate with different devices. In other words, the command-queue 104 permits the proper execution in the cases of: (1) one client to many devices, (2) many clients to one device, and (3) many clients to many devices.

The present inventor came to the realization that the digital command stations provided by the different vendors 40 have at least three different techniques for communicating with the digital decoders of the model railroad set. The first technique, generally referred to as a transaction (one or more operations), is a synchronous communication where a command is transmitted, executed, and a response is received 45 therefrom prior to the transmission of the next sequentially received command. The DCS may execute multiple commands in this transaction. The second technique is a cache with out of order execution where a command is executed and a response received therefrom prior to the execution of 50 the next command, but the order of execution is not necessarily the same as the order that the commands were provided to the command station. The third technique is a local-area-network model where the commands are transmitted and received simultaneously. In the LAN model there 55 is no requirement to wait until a response is received for a particular command prior to sending the next command. Accordingly, the LAN model may result in many commands being transmitted by the command station that have yet to be executed. In addition, some digital command stations use 60 two or more of these techniques.

With all these different techniques used to communicate with the model railroad set and the system 10 providing an interface for each different type of command station, there exists a need for the capability of matching up the responses 65 from each of the different types of command stations with the particular command issued for record keeping purposes.

Without matching up the responses from the command stations, the databases can not be updated properly.

Validation functionality is included within the external device control logic 114 to accommodate all of the different types of command stations. Referring to FIG. 3, an external command processor 200 receives the validated command from the synchronous command processor 110. The external command processor 200 determines which device the command should be directed to, the particular type of command it is, and builds state information for the command. The state information includes, for example, the address, type, port, variables, and type of commands to be sent out. In other words, the state information includes a command set for a particular device on a particular port device. In addition, a copy of the original command is maintained for verification purposes. The constructed command is forwarded to the command sender 202 which is another queue, and preferably a circular queue. The command sender 202 receives the command and transmits commands within its queue in a repetitive nature until the command is removed from its queue. A command response processor 204 receives all the commands from the command stations and passes the commands to the validation function 206. The validation function 206 compares the received command against potential commands that are in the queue of the command sender 202 that could potentially provide such a result. The validation function 206 determines one of four potential results from the comparison. First, the results could be simply bad data that is discarded. Second, the results could be partially executed commands which are likewise normally discarded. Third, the results could be valid responses but not relevant to any command sent. Such a case could result from the operator manually changing the state of devices on the model railroad or from another external device, assuming a shared interface to the DCS. Accordingly, the results are validated and passed to the result processor 210. Fourth, the results could be valid responses relevant to a command sent. The corresponding command is removed from the command sender 202 and the results passed to the result processor 210. The commands in the queue of the command sender 202, as a result of the validation process 206, are retransmitted a predetermined number of times, then if error still occurs the digital command station is reset, which if the error still persists then the command is removed and the operator is notified of the error.

APPLICATION PROGRAMMING INTERFACE

Train ToolsTM Interface Description Building your own visual interface to a model railroad Copyright 1992-1998 KAM Industries. Computer Dispatcher, Engine Commander, The Conductor, Train Server, and Train Tools are Trademarks of KAM Industries, all Rights Reserved. Questions concerning the product can be EMAILED to: traintools@kam.rain.com You can also mail questions to: KAM Industries 2373 NW 185th Avenue Suite 416

Hillsboro, Oregon 97124 FAX - (503) 291-1221

Table of conte

- OVERVIEW 1.1 System Architecture
- TÚTORIAL
- Visual BASIC Throttle Example Application
- Visual BASIC Throttle Example Source Code
- IDL COMMAND REFERENCE

9

-continued

10 -continued

APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE 5 3.9 3.1 Introduction Commands to configure the command station 3.2 3.3 Data Types munication port KamPortPutConfig Commands to access the server configuration variable KamPortGetConfig database KamPortGetName KamCVGetValue KamCVPutValue KamPortPutMapController KamCVGetEnable 10 KamPortGetMaxLogPorts KamPortGetMaxPhysical KamCVPutEnable KamCVGetName Commands that control command flow to the command 3.10 KamCVGetMinRegister station KamCVGetMaxRegister KamCmdConnect Commands to program configuration variables KamCmdDisConnect KamProgram KamCmdCommand 3.11 Cab Control Commands KamProgramGetMode KamProgramGetStatus KamCabGetMessage KamCabPutMessage KamCabGetCabAddı KamProgramReadCV KamProgramCV KamProgramReadDecoderToDataBase KamCabPutAddrToCab KamProgramDecoderFromDataBase Commands to control all decoder types 20 3.12 Miscellaneous Commands KamMiscGetErrorMsg 3.5 KamDecoderGetMaxModels KamMiscGetClockTime KamDecoderGetModelName KamDecoderSetModelToObj KamMiscPutClockTime KamMiscGetInterfaceVersion KamDecoderGetMaxAddress KamMiscSaveData KamDecoderChangeOldNewAddr KamDecoderMovePort KamMiscGetControllerName 25 KamMiscGetControllerNameAtPort KamDecoderGetPort KamMiscGetCommandStationValue KamDecoderCheckAddrInUse KamMiscSetCommandStationValue KamDecoderGetModelFromObj KamMiscGetCommandStationIndex KamDecoderGetModelFacility KamMiscMaxControllerID KamDecoderGetObjCount KamDecoderGetObjAtIndex KamMiscGetControllerFacility OVERVIEW 30 I. This document is divided into two sections, the KamDecoderPutAdd Tutorial, and the IDL Command Reference. The tutorial shows the complete code for a simple Visual BASIC program KamDecoderPutDel KamDecoderGetMfgName KamDecoderGetPowerMode that controls all the major functions of a locomotive KamDecoderGetMaxSpeed This program makes use of many of the commands described in the reference section. The IDL Command Reference Commands to control locomotive decoders 3.6 describes each command in detail. KamEngGetSpeed KamEngPutSpeed KamEngGetSpeedSteps TITORIAI. A. Visual BASIC Throttle Example Application Visual BASIC Infolie Example Application The following application is created using the Visual BASIC source code in the next section. It controls all major locomotive functions such as speed, direction, and auxiliary functions. A. Visual BASIC Throttle Example Source Code 'Copyright 1998, KAM Industries. All rights reserved. KamEngPutSpeedSteps KamEngGetFunction KamEngPutFunction KamEngGetFunctionMax KamEngGetName KamEngPutName KamEngGetFunctionName This is a demonstration program showing the integration of VisualBasic and Train Server(tm) interface. You may use this application for non KamEngPutFunctionName KamEngGetConsistMax 45 KamEngPutConsistParent KamEngPutConsistChild KamEngPutConsistRemoveObj commercial usage '\$Date: \$ Commands to control accessory decoders KamAccGetFunction '\$ Author: \$ KamAccGetFunctionAll \$Revision: \$ KamAccPutFunction 50 '\$Log: \$ KamAccPutFunctionAll Engine Commander, Computer Dispatcher, Train Server, Train Tools, The Conductor and kamind are registered KamAccGetFunctionMax KamAccGetName Trademarks of KAM Industries. All rights reserved. KamAccPutName KamAccGetFunctionName This first command adds the reference to the Train KamAccPutFunctionName ServerT Interface object Dim EngCmd As New EngComIfc 55 KamAccRegFeedback KamAccRegFeedbackAll Engine Commander uses the term Ports, Devices and KamAccDelFeedback KamAccDelFeedbackAll Controllers Ports -> These are logical ids where Decoders are Commands to control the command station assigned to. Train ServerT Interface supports a KamOprPutTumOnStation KamOprPutStartStation limited number of logical ports. You can also think 60 of ports as mapping to a command station type. This allows you to move decoders between command station KamOprPutClearStation KamOprPutStopStation KamOprPutPowerOn without losing any information about the decoder KamOprPutPowerOff Devices -> These are communications channels KamOprPutHardReset configured in your computer. 65 You may have a single device (com1) or multiple KamOprPutEmergencyStop KamOprGetStationStatus

11

12

		-co	ontinued		-co	ntinued
	Al	PPLICATION PRO	GRAMMING INTERFACE		APPLICATION PRO	GRAMMING INTERFACE
. (0	COM 1	- COM8, LPT1, C	other). You are required to	5	LENZ_1x	2 // Lenz serial support module
			ccess a command station.		LENZ_2x	3 // Lenz serial support module 4 // Digitrax direct drive
			max id (FYI; devices do serial channel. Always		'DIGIT_DT200	support using DT200
			ce before you use it as		' DIGIT_DCS100	5 // Digitrax direct drive
			per of devices supported.		1344 CTED CEDIEC	support using DCS100
	he Cor		hysical(IMaxPhysical, ISerial,	10	' MASTERSERIES	6 // North Coast engineering master Series
			hat IMaxPhysical =		'SYSTEMONE	7 // System One
18	Serial +	Parallel + lOther			' RAMFIX	8 // RAMFixx system
: .	ontroll	er - These are com	mand the command station		' DYNATROL ' Northcoast binary	9 // Dynatrol system 10 // North Coast binary
		NZ, Digitrax		15	' SERIAL	11 // NMRA Serial
			klin It is recommend		'EASYDCC	interface 12 // NMRA Serial interface
	nat you ise it.	check the comman	nd station ID before you		MRK6050	13 // 6050 Marklin interface
1						(AC and DC)
' E	Errors		s return an error status. If is non zero, then the		'MRK6023	14 // 6023 Marklin hybrid interface (AC)
			guments are invalid. In	20	'ZTC	15 // ZTC Systems ltd
•			ero errors means command was		' DIGIT_PR1	16 // Digitrax direct drive
			To get the error message, II KamMiscErrorMessage and		'DIRECT	support using PR1 17 // Direct drive interface
		supply the erro			binder	routine
•		•••		25	***********	********
			u will need to perform a ogical reference), Device	23	iLogicalPort = 1 'Select Lo	communications
' (physica	d communications	channel) and a Controller		iController = 1 'Select cont	
			program to work. All		:CP 0.1 COM1	above.
		ces uses the logical for access.	device as the reference		iComPort = 0 'use COM1;	0 means com1 (Digitrax must use Com1 or Com2)
, ,		or access.		30	'Digitrax Baud rate	requires 16.4K!
			ect reference. To use an		'Most COM ports al	
			address to the command PutAdd One of the return		'support 16.4K. Che 'manufacture of you	
· •	values f	rom this operation	is an object reference		for the baud rate. K	Leep in mind that
, ti	hat is t	sed for control.			'Dumb com cards w	rith serial port m4 can only support
, ,	We nee	d certain variables	as global objects; since	35	'2 com ports (like c	
' t	he info	rmation is being us	ed multiple times		or com3/com4)	
		ort, iController, iCo iPortParity, iPortSt			'If you change the c 'forget to change the	
		tchdog, iPortFlow,			match the comman	
			coderClass As Integer,	40	'user manual for det	ails
		erType As Intege roller As Long	r		'0: // Baud rate is 3	300
			Physical As Long, IMaxSerial		' 1: // Baud rate is 1	200
	As Lon	g, lMaxParallel As	Long		' 2: // Baud rate is 2 ' 3: // Baud rate is 4	
Form lo	oad fun	ction			'4: // Baud rate is 9	
		nitial buttons		45	'5: // Baud rate is 1	
'- Set he	interfa	ace information	****		' 6: // Baud rate is 1 ' 7: // Baud rate is 1	
Private :	Sub Fo	rm_load()			iPortRate = 4	
I			om As String, strCntrl As			-4 -> no, odd, even, mark,
ī		String mor As Integer		50	space iPortParity = 0	
•	Get the	interface version i	nformation		' Stop bits 0,1,2	-> 1, 1.5, 2
		onState (False)	cGetInterfaceVersion(strVer)		iPortStop = 0 iPortRetrans = 10	
		or) Then	COethite hace version(sir ver)		iPortWatchdog = 20)48
	•		erver not loaded. Check		iPortFlow = 0	77 4 015
		DCOM-95")) iLogicalPort = 0		55	iPortData = 1	7 Bits, 1-> 8 bits
		LogPort.Caption =			Display the port and contr	
		ComPort.Caption =				GetMaxLogPorts(lMaxLogical)
1	Else	Controller.Caption	= "Unknown"		lError = EngCmd.KamPort	GetMaxPhysical(lMaxPhysical, faxParallel)
		MsgBox (("Simula	tion(COM1) Train Server " &	60	' Get the port name and do	some checking
		strVer))	*******	00	iError = EngCmd.KamPort SetError (iError)	GetName(iComPort, strCom)
			rmation; Only need to) Then MsgBox ("Com port
		change these	values to use a different		our of range")	÷ .
		controller	*******		iError = EngCmd.KamMisc	GetControllerName(iController,
		UNKNOWN	0 // Unknown control type	65	strCntrl)	•
		SIMULAT	1 // Interface simulator		If (iLogicalPort > lMaxLo	gical) Then MsgBox

13

-continued

14

```
APPLICATION PROGRAMMING INTERFACE
                                                                                                          APPLICATION PROGRAMMING INTERFACE
                                                                                                   iError = EngCmd.KamPortPutConfig(iLogicalPort, 4
("Logical port out of range")
                                                                                                   iPortWatchdog, 0) 'setting PORT_WATCHDOG iError = EngCmd.KamPortPutConfig(iLogicalPort, 5,
                 SetError (iError)
     End If
                                                                                                   iPortFlow, 0) 'setting PORT_FLOW
iError = EngCmd.KamPortPutConfig(iLogicalPort, 6,
iPortData, 0) 'setting PORT_DATABITS
        'Display values in Throttle.
        LogPort.Caption = iLogicalPort
        ComPort.Caption = strCom
                                                                                       10 'We need to set the appropriate debug mode for display...
'this command can only be sent if the following is true
        Controller.Caption = strCntrl
End Sub
                                                                                             -Controller is not connected
Send Command
                                                                                             -port has not been mapped
                                                                                             -Not share ware version of application (Shareware
'Note:
                                                                                                   always set to 130)
        Please follow the command order. Order is important
                                                                                            Write Display Log
File Win Level
        for the application to work!
                                                                                                                        Debug
                                                                                                                       Value
                                                                                             1 + 2 + 4 = 7
                                                                                                                            -> LEVEL1 -- put packets into
Private Sub Command_Click()
        'Send the command from the interface to the command
                                                                                                   queues
        station, use the engineObject
                                                                                            1 + 2 + 8 = 11
                                                                                                                            -> LEVEL2 -- Status messages
        Dim iError, iSpeed As Integer
If Not Connect.Enabled Then
                                                                                                   send to window
                                                                                                                            -> LEVEL 3 --
                                                                                               +2+16=19
                 TrainTools interface is a caching interface.
                                                                                            1 + 2 + 32 = 35
                                                                                                                            -> LEVEL4 -- All system
                 This means that you need to set up the CV's or other operations first; then execute the
                                                                                                   semaphores/critical sections
                                                                                            1 + 2 + 64 = 67
                                                                                                                            -> LEVEL5 -- detailed
                                                                                            debugging information
1 + 2 + 128 = 0 131
                 command.
                                                                                                                            -> COMMONLY -- Read comm write
                 iSpeed - Speed.Text
                 iError =
        EngCmd.KamEngPutFunction(lEngineObject, 0, F0.Value)
                                                                                             You probably only want to use values of 130. This will
                 iError -
                 EngCmd.KamEngPutFunction(IEngineObject, 1,
                                                                                            give you a display what is read or written to the
                 F1. Value)
                                                                                            controller. If you want to write the information to
                                                                                            'disk, use 131. The other information is not valid for
                 iError =
                 EngCmd.KamEngPutFunction(lEngineObject, 2,
                                                                                            end users
                                                                                                               This does effect the performance of you
                 F2. Value)
                                                                                            ' Note: 1.
                                                                                                               system; 130 is a save value for debug
                 iError =
                                                                                                               display. Always set the key to 1, a value of 0 will disable debug
                 EngCmd.KamEngPutFunction(lEngineObject, 3,
                 F3. Value)
iError = EngCmd.KamEngPutSpeed(lEngineObject,
                                                                                                               The Digitrax control codes displayed are
                 iSpeed, Direction. Value)
                                                                                                               encrypted. The information that you
                 If iError = 0 Then iError =
                                                                                                               determine from the control codes is that
                                                                                       35
                                                                                                               information is sent (S) and a response is
                 EngCmd.KamCmdCommand(lEngineObject)
                 SetError (iError)
                                                                                                               received (R)
              End If
End Sub
                                                                                            iDebugMode = 130
                                                                                            iValue = Value. Text' Display value for reference
iError = EngCmd. KamPortPutConfig(iLogicalPort, 7, iDebug,
iValue)' setting PORT_DEBUG
         ***********
'Connect Controller
                                                                                            'Now map the Logical Port, Physical device, Command
station and Controller
Private Sub Connect_Click()
        Dim iError As Integer
         "These are the index values for setting up the port
                                                                                            iError = EngCmd.KamPortPutMapController(iLogicalPort,
                                                                                            iController, iComPort)
iError = EngCmd.KamCmdConnect(iLogicalPort)
          PORT_RETRANS
                                            0 // Retrans index
          PORT_RATE
                                                                                            iError = EngCmd.KamOprPutTurnOnStation(iLogicalPort)
                                             1 // Retrans index
                                                                                            If (iError) Then
SetButtonState (False)
          PORT_PARITY
                                             2 // Retrans index
          PORT_STOP
                                             3 // Retrans index
         PORT_WATCHDOG
PORT_FLOW
PORT_DATABITS
                                             4 // Retrans index
                                                                                                 Else
                                                                                                    SetButtonState (True)
                                            5 // Retrans index
                                                                                                  End If
                                             6 // Retrans index
          PORT_DEBUG
                                             7 // Retrans index
                                                                                       50 SetError (iError) 'Displays the error message and error
          PORT_PARALLEL 8 // Retrans index
These are the index values for setting up the
                                                                                                    number
                                                                                            End Sub
                 port for use
         'PORT_RETRANS
'PORT_RATE
                                             0 // Retrans index
                                                                                            'Set the address button
                                             1 // Retrans index
          PORT_PARTTY
                                             2 // Retrans index
                                                                                       55 Private Sub DCCAddr_Click()
         PORT_STOP
PORT_WATCHDOG
                                            3 // Retrans index
4 // Retrans index
                                                                                                    Dim iAddr, iStatus As Integer
'All addresses must be match to a logical port to
          PORT_FLOW
                                             5 // Retrans index
                                                                                                    PORT_DATABITS
PORT_DEBUG
                                             6 // Retrans index
7 // Retrans index
                                                                                                                                  ' Set the decoder type to an NMRA
          PORT_PARALLEL
                                             8 // Retrans index
         'PORT_PARALIEL 8 // Retrans index
iError = EngCmd.KamPortPutConfig(iLogicalPort, 0,
iPortRetrans, 0) 'setting PORT_RETRANS
iError = EngCmd.KamPortPutConfig(iLogicalPort, 1
iPortRate, 0) 'setting PORT_RATE
iError = EnqCmd.KamPortPutConfig(iLogicalPort, 2,
iPortParity, 0) 'setting PORT_PARITY
iError = EngCmd.KamPortPutConfig(iLogicalPort, 3)
iError = EngCmd.KamPortPutConfig(iLogicalPort, 3)
                                                                                       60
                                                                                                     Engine and Accessory
                                                                                                     'Once we make a connection, we use the lEngineObject
                                                                                                     'as the reference object to send control information
                                                                                                     If (Address.Text > 1) Then
                                                                                                          iStatus = EngCmd.KamDecoderPutAdd(Address.Text,
iLogicalPort, iLogicalPort, 0,
                                                                                        65
                 iPortStop, 0) 'setting PORT_STOP
                                                                                                                iDecoderType, lEngineObject)
```

16

```
-continued
                                                                                                                  -continued
             APPLICATION PROGRAMMING INTERFACE
                                                                                               APPLICATION PROGRAMMING INTERFACE
       SetError (iStatus)
                                                                                  Private Sub ONCmd_Click()
       If(IEngineObject) Then
                                                                                          Dim iError As Integer
            Command.Enabled = True 'turn on the control
                                                                                          iError = EngCmd.KamOprPutPowerOn(iLogicalPort)
            (send) button
            Throttle. Enabled = True Turn on the throttle
                                                                                          SetError (iError)
                                                                                  End Sub
            MsgBox ("Address not set, check error message")
                                                                              10
                                                                                  Throttle slider control
            End If
                                                                                  Private Sub Throttle_Click()
            MsgBox ("Address must be greater then 0 and
                                                                                          If (lEngineObject) Then
                 less then 128")
                                                                                              If (Throttle. Value > 0) Then
                                                                                                 Speed.Text = Throttle.Value
End If
End Sub
                                                                              15
Disconenct button
                                                                                               End If
                                                                                  End Sub
                                                                                          IDL COMMAND REFERENCE
Private Sub Disconnect_Click()
                                                                                          A. Introduction
       Dim iError As Integer
                                                                                               This document describes the IDL interface to
       iError = EngCmd.KamCmdDisConnect(iLogicalPort)
       SetError (iError)
                                                                                  the KAM Industries Engine Commander Train Server. The
                                                                                  Train Server DCOM server may reside locally or on a
network node This server handles all the background
       SetButtonState (False)
End Sub
                                                                                   details of controlling your railroad. You write simple,
'Display error message
                                                                                  front end programs in a variety of languages such as BASIC, Java, or C++ to provide the visual interface to the user while the server handles the details of
Private Sub SetError(iError As Integer)
                                                                                  communicating with the command station, etc.

A. Data Types
       Dim szError As String
       Dim iStatus
                                                                                   Data is passed to and from the IDL interface using a
        This shows how to retrieve a sample error message
                                                                                  several primitive data types. Arrays of these simple types are also used. The exact type passed to and from
       from the interface for the status received.
       iStatus = EngCmd.KamMiscGetErrorMsg(iError, szError)
                                                                                  your program depends on the programming language your are
       ErrorMsg.Caption = szError
                                                                                  using.
The following primitive data types are used:
        Result.Caption = Str(iStatus)
End Sub
                                                                                   IDL Type BASIC Type C++ Type Java Type
                                                                                                                                      Description
                                                                                  short
'Set the Form button state
                                                                                               short
                                                                                                             short
                                                                                                                          short
                                                                                                                                      Short signed integer
                                                                                                                                      Signed integer
                                                                                               int
                                                                                                              int
                                                                                                                          int
                                                                                                                                      Text string
Unsigned 32 bit value
                                                                                  BSTR
                                                                                               BSTR
                                                                                                              BSTR
                                                                                                                          BSTR
Private Sub SetButtonState(iState As Boolean)
                                                                              35
                                                                                  long long long long
Name ID CV Range Valid CV's Functions
        'We set the state of the buttons; either connected
                                                                                                                                     Address Range
        or disconnected
       If (iState) Then
            Connect.Enabled = False
Disconnect.Enabled = True
                                                                                                             None
                                                                                   NMRA Compatible 0
                                                                                                                       None
                                                                                                                                      1-99
                                                                                                                                      1-127
                                                                                                             1-8
                                                                                                                       1-8
                                                                                   Baseline
             ONCmd.Enabled = True
                                                                                   Extended
                                                                                                             1-106
                                                                                                                       1-9, 17, 18, 19, 23, 24, 29, 30,
             OffCmd.Enabled = True
                                                                                  40 66-05
                                                                                                    9
                                                                                                              1-10239
                                                                                                                              14,28,128
9 1-1
                                                                                                                       1-106
                                                                                                                                      1-10239
                                                                                                                                                  14,28,128
             DCCAddr.Enabled = True
                                                                                   All Mobile
                                                                                                        3
                                                                                                             1-106
                                                                                                 CV Range
                                                                                                                Valid CV's
             UpDownAddress.Enabled = True
                                                                                   Name ID
                                                                                                                                 Functions Address Range
       'Now we check to see if the Engine Address has been
                                                                                                                513-593
                                                                                                                               513-593 8 0-511
                                                                                                                              513-1024 8
       'set; if it has we enable the send button If (lEngineObject > 0) Then
                                                                                                                513-1024
                                                                                   All Stationary
                                                                                   A long /DecoderObject/D value is returned by the
             Command.Enabled = True
                                                                                  KamDecoderPutAdd call if the decoder is successfully
                                                                                   registered with the server. This unique opaque ID should
             Throttle.Enabled = True
                                                                                   be used for all subsequent calls to reference this
          Else
             Command.Enabled = False
                                                                                   decoder.
             Throttle.Enabled = False
                                                                                          Commands to access the server configuration variable
          End If
                                                                                               This section describes the commands that access
                                                                                   the server configuration variables (CV) database. These
             Connect.Enabled = True
             Disconnect.Enabled = False
                                                                                   CVs are stored in the decoder and control many of its
                                                                                  characteristics such as its address. For efficiency, a copy of each CV value is also stored in the server
             Command Enabled = False
             ONCmd.Enabled = False
                                                                                  database. Commands such as KamCVGetValue and KamCVPutValue communicate only with the server, not the
             OffCmd.Enabled = False
             DCCAddr.Enabled = False
             UpDownAddress.Enabled = False
                                                                                   actual decoder. You then use the programming commands in
             Throttle.Enabled = False
                                                                                   the next section to transfer CVs to and from the decoder.
             End If
                                                                                   0KamCVGetValue
                                                                                                                          Direction
End Sub
                                                                                   Parameter List
                                                                                                                Range
                                                                                   Parameter List -7F | IDecoderObjectID long 1 | 1-1024 2
                                                                                                       Type
                                                                                                                                       Description
                                                                                                                          In
                                                                                                                                       Decoder object ID
Power Off function
                                                                                                                                       CV register
                                                                                                                          Īn
                                                                                                                                       Pointer to CV value
                                                                                   pCV Value
Private Sub OffCmd_Click()
                                                                                          Opaque object ID handle returned by
                                                                                   KamDecoderPutAdd.
        Dim iError As Intege
        iError = EngCmd.KamOprPutPowerOff(iLogicalPort)
                                                                                          Range is 1-1024. Maximum CV for this decoder is
        SetError (iError)
                                                                                   given by KamCVGetMaxRegister.
                                                                                          CV Value pointed to has a range of 0 to 255.
End Sub
                                                                                   Return Value
                                                                                                       Туре
                                                                                                                      Range
                                                                                                                                       Description
Power On function
                                                                                   iError short
                                                                                                                     Error flag
```

17

18

-continued -continued APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE

AFF LICATION FROOKAMMING INTERFACE		AFFLICATION FROOKAMMING INTERFACE			
1 iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamCVGetValue takes the	Reco	VNameString to the name of the CV as defined in NMRA mmended Practice RP 9.2.2.			
decoder object ID and configuration variable (CV) number		1CVGetMinRegister			
as parameters. It sets the memory pointed to by pCVValue		neter List Type Range Direction Description			
to the value of the server copy of the configuration variable.		oderObjectID long 1 In Decoder object ID Register int * 2 Out Pointer to min CV			
0KamCVPutValue	10 p.viii	register number			
Parameter List Type Range Direction Description	1	Opaque object ID handle returned by			
lDecoderObjectID long 1 In Decoder object ID	Kam	DecoderPutAdd.			
iCVRegint 1-1024 2 In CV register	2	Normally 1-1024. 0 on error or if decoder does not			
iCVValue int 0-255 In CV value		ort CVs.			
1 Opaque object ID handle returned by	1977	n Value Type Range Description r short 1 Error flag			
KamDecoderPutAdd. 2 Maximum CV is 1024. Maximum CV for this decoder is	15 iErro	r short 1 Error flag iError = 0 for success. Nonzero is an error number			
given by KamCVGetMaxRegister.		KamMiscGetErrorMsg).			
Return Value Type Range Description		CVGetMinRegister takes a decoder object ID as a			
iError short 1 Error flag		neter. It sets the memory pointed to by pMinRegister			
iError = 0 for success. Nonzero is an error number		e minimum possible CV register number for the			
(see KamMiscGetErrorMsg).		fied decoder. nCVGetMaxRegister			
KamCVPutValue takes the decoder object ID, configuration variable (CV) number, and a new CV value as parameters.		neter List Type Range Direction Description			
It sets the server copy of the specified decoder CV to		oderObjectID long 1 In Decoder object ID			
iCVValue.		Register int * 2 Out Pointer to max CV			
0KamCVGetEnable	regis	er number			
Parameter List Type Range Direction Description	25 7	Opaque object ID handle returned by			
lDecoderObjectID long 1 In Decoder object ID iCVRegint 1–1024 2 In CV number		DecoderPutAdd.			
iCVRegint 1-1024 2 In CV number pEnable int * 3 Out Pointer to CV bit mask	2 supp	Normally 1–1024. 0 on error or if decoder does not out CVs.			
1 Opaque object ID handle returned by		n Value Type Range Description			
KamDecoderPutAdd.		r short 1 Error flag			
2 Maximum CV is 1024. Maximum CV for this decoder is	1	iError = 0 for success. Nonzero is an error number			
given by KamCVGetMaxRegister.		KamMiscGetErrorMsg).			
3 0x0001 - SET_CV_INUSE 0x0002 - SET_CV_READ_DIRTY 0x0004 - SET_CV_WRITE_DIRTY 0x0008 -		CVGetMaxRegister takes a decoder object ID as a			
0x0004 - SET_CV_WRITE_DIRTY 0x0008 - SET_CV_ERROR_READ		neter. It sets the memory pointed to by pMaxRegister maximum possible CV register number for the			
0x0010 - SET_CV_ERROR_WRITE		fied decoder.			
Return Value Type Range Description	A.	Commands to program configuration variables			
iError short 1 Error flag	35	This section describes the commands read and			
1 iError = 0 for success. Nonzero is an error number	write	decoder configuration variables (CVs). You should			
(see KamMiscGetErrorMsg). KamCVGetEnable takes the		initially transfer a copy of the decoder CVs to the			
decoder object ID, configuration variable (CV) number, and a pointer to store the enable flag as parameters. It		r using the KamProgramReadDecoderToDataBase command. can then read and modify this server copy of the CVs.			
sets the location pointed to by pEnable.		ty, you can program one or more CVs into the decoder			
0KamCVPutEnable		the KamProgramCV or KamProgramDecoderFromDataBase			
Parameter List Type Range Direction Description	come	nand. Not that you must first enter programining mode			
iDecoderObjectID long 1 In Decoder object ID		suing the KamProgram command before any programming			
iCVRegint 1-1024 2 In CV number iEnableint 3 In CV bit mask		e done. 1Program			
1 Opaque object ID handle returned by		neter List Type Range Direction Description			
KamDecoderPutAdd.	lDece	oderObjectID long 1 In Decoder object ID			
2 Maximum CV is 1024. Maximum CV for this decoder is	45 iProg	LogPort int 1-65535 2 In Logical			
given by KamCVGetMaxRegister.		programming			
3 0x0001 - SET_CV_INUSE 0x0002 - SET_CV_READ_DIRTY 0x0004 - SET_CV_WRITE_DIRTY 0x0008 -	iProg	port ID Mode int 3 In Programming mode			
SET_CV_ERROR_READ	1	Opaque object ID handle returned by			
0x0010 - SET_CV_ERROR_WRITE	Kam	DecoderPutAdd.			
Return Value Type Range Description	50 2	Maximum value for this server given by			
iError short 1 Error flag		PortGetMaxLogPorts.			
1 iError = 0 for success. Nonzero is an error number	3	0 - PROGRAM_MODE_NONE 1 - PROGRAM_MODE_ADDRESS			
(see KamMiscGetErrorMsg). KamCVPutEnable takes the decoder object ID, configuration		2 - PROGRAM_MODE_REGISTER			
variable (CV) number, and a new enable state as		3 - PROGRAM_MODE_PAGE			
parameters. It sets the server copy of the CV bit mask	55	4 - PROGRAM_MODE_DIRECT			
to iEnable.		5 - DCODE_PRGMODE_OPS_SHORT			
0KamCVGetName	ъ.	6 - PROGRAM_MODE_OPS_LONG			
Parameter List Type Range Direction Description iCV int 1-1024 In CV number		n Value Type Range Description r short 1 Error flag			
pbsCVNameString BSTR * 1 Out Pointer to CV	1	iError = 0 for success. Nonzero is an error number			
name string	(see	KamMiscGetErrorMsg)			
1 Exact return type depends on language. It is		Program take the decoder object ID, logical			
Cstring * for C++. Empty string on error.		amming port ID, and programming mode as parameters.			
Return Value Type Range Description iError short 1 Error flag		anges the command station mode from normal operation OGRAM_MODE_NONE) to the specified programming mode.			
1 iError = 0 for success. Nonzero is an error number		in programming modes, any number of programming			
(see KamMiscGetErrorMsg).		nands may be called. When done, you must call			
KamCVGetName takes a configuration variable (CV) number	65 Kam	Program with a parameter of PROGRAM_MODE_NONE to			
as a parameter. It sets the memory pointed to by	retur	n to normal operation.			

(see KamMiscGetErrorMsg).

US 6,530,329 B2

19

20

of the decoder at address lDecoderObjectID to the type

-continued -continued APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE $0 \\ Kam Program Get \\ Mode \\$ KamProgramCV takes the decoder object ID, configuration variable (CV) number, and a new CV value as parameters. Parameter List Type Range Direction Description It programs (writes) a single decoder CV using the 1DecoderObjectID long Decoder object ID 1-65535 specified value as source data. OKamProgramReadDecoderToDataBase int 2 Logical iProgLogPort programming port ID Type Range Direction lDecoderObjectID long int * 3 Ont Programming mode 1 Ĭn Decoder object ID piProgMode Opaque object ID handle returned by Opaque object ID handle returned by KamDecoderPutAdd. KamDecoderPutAdd. Range Error flag Maximum value for this server given by Return Value Type Description KamPortGetMaxLogPorts. 3 0 - PROGRAM_MODE_NONE iError short 1 iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamProgramReadDecoderToDataBase takes the decoder object 1 - PROGRAM_MODE_ADDRESS 2 - PROGRAM_MODE_REGISTER 3 - PROGRAM_MODE_PAGE ID as a parameter. It reads all enabled CV values from 4 - PROGRAM_MODE_DIRECT the decoder and stores them in the server database. 5 - DOODE PROMODE OPS SHORT 0KamProgramDecoderFromDataBase 6 - PROGRAM_MODE_OPS_LONG Type Range Direction IDecoderObjectID long 1 In Opaque object ID handle returned by Return Value Туре Range Description Decoder object ID iError short 1 Error flag Description KamDecoderPutAdd. iError = 0 for success. Nonzero is an error number Range Error flag (see KamMiscGetErrorMsg). Return Value Туре Description iError short KamProgramGetMode take the decoder object ID, logical iError = 0 for success. Nonzero is an error number programming port ID, and pointer to a place to store (see KamMiscGetErrorMsg). KamProgramDecoderFromDataBase takes the decoder object ID the programming mode as parameters. It sets the memory pointed to by piProgMode to the present programming mode. 0KamProgramGetStatus as a parameter. It programs (writes) all enabled decoder Parameter List Type Range Direction Description CV values using the server copy of the CVs as source IDecoderObjectID long Decoder object ID data. In CV number Commands to control all decoder types iCVRegint 0-1024 piCVAllStatus This section describes the commands that all decoder types. These commands do things such getting the int * Out Or'd decoder programming status maximum address a given type of decoder supports, adding Opaque object ID handle returned by KamDecoderPutAdd. decoders to the database, etc 0KamDecoderGetMaxModels 0 returns OR'd value for all CVs. Other values Parameter List Type Range Direction Description return status tor just that CV. 0x0001 - SET_CV_INUSE 0x0002 - SET_CV_READ_DIRTY piMaxModels Out Pointer to Max model ID 35 Normally 1-65535. 0 on error. 0x0004 - SET_CV_WRITE_DIRTY Return Value 0x0008 - SET_CV_ERROR_READ 0x0010 - SET_CV_ERROR_WRITE Range Туре Description Error flag iError short Return Value iError = 0 for success. Nonzero is an error number Type Range Description (see KamMiscGetErrorMsg). KamDecoderGetMaxModels takes no parameters. It sets the iError short Error flag iError = 0 for success. Nonzero is an error number memory pointed to by piMaxModels to the maximum decoder (see KamMiscGetErrorMsg). type ID. 0KamDecoderGetModelName KamProgramGetStatus take the decoder object ID and pointer to a place to store the OR'd decoder programming status as parameters. It sets the memory pointed to by Parameter List Type 1-65535 Range Direction piProgMode to the present programming mode. 0KamProgramReadCV iModel int In Decoder type ID pbsModelName BSTR * Decoder name Out Type Range Direction Description string lDecoderObjectID long 1 In Decoder object ID Maximum value for this server given by CV number KamDecoderGetMaxModels. iCVRegint In Opaque object ID handle returned by Exact return type depends on language. It is KamDecoderPutAdd. Cstring * for C++. Empty string on error. Return Value Type Range Maximum CV is 1024. Maximum CV for this decoder is Туре Description given by KamCVGetMaxRegister. Error flag 1 iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamPortGetModelName takes a Return Value Type Range Error flag Description iError short iError = 0 for success. Nonzero is an error number decoder type ID and a pointer to a string as parameters (see KamMiscGetErrorMsg). KamProgramCV takes the decoder object ID, configuration It sets the memory pointed to by pbsModelName to a BSTR containing the decoder name. variable (CV) number as parameters. It reads the 0KamDecoderSetModelToObj Type Range Direction int 1 In specified CV variable value to the server database. 0KamProgramCV Parameter List Description Decoder model ID Direction Description IDecoderObjectID long In Decoder object ID Parameter List Type Range Maximum value for this server given by iDecoderObjectID long 1 In Decoder object ID iCVRegint CV number KamDecoderGetMaxModels. In Opaque object ID handle returned by KamDecoderPutAdd. iCVValue 0-255 In CV value Opaque object ID handle returned by KamDecoderPutAdd. Return Value Туре Range Description 2 Maximum CV is 1024. Maximum CV for this decoder is given by KamCVGetMaxRegister. Return Value Type Range Description iError short Error flag iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). Error flag iError short KamDecoderSetModelToObj takes a decoder ID and decoder iError = 0 for success. Nonzero is an error number object ID as parameters. It sets the decoder model type

21

-continued

22 -continued

APPLICATION PROGRAMMING INTERFACE		APPLICATION PROGRAMMING INTERFACE
specified by iModel.	5	KamPortGetMaxLogPorts.
0KamDecoderGetMaxAddress		3 1 - DECODER_ENGINE_TYPE,
Parameter List Type Range Direction Description		2 - DECODER_SWITCH_TYPE,
iModel int 1 In Decoder type ID piMaxAddress int * 2 Out Maximum decoder		3 - DECODER_SENSOR_TYPE. Return Value Type Range Description
piMaxAddress int * 2 Out Maximum decoder address		iError short 1 Error flag
1 Maximum value for this server given by	10	
KamDecoderGetMaxModels		use. Nonzero is an error number (see
2 Model dependent. 0 returned on error.		KamMiscGetErrorMsg). IDS_ERR_ADDRESSEXIST returned if
Return Value Type Range Description		call succeeded but the address exists.
iError short 1 Error flag 1 iError = 0 for success. Nonzero is an error number		KamDecoderCheckAddrInUse takes a decoder address, logical port, and decoder class as parameters. It returns zero
(see KamMiscGetErrorMsg).	15	10 1 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
KamDecoderGetMaxAddress takes a decoder type ID and a	13	IDS_ERR_ADDRESSEXIST if the call succeeds but the address
pointer to store the maximum address as parameters. It		already exists. It will return the appropriate non zero
sets the memory pointed to by piMaxAddress to the maximum		error number if the calls fails.
address supported by the specified decoder.		0KamDecoderGetModelFromObj Parameter List Type Range Direction Description
0KamDecoderChangeOldNewAddr Parameter List Type Range Direction Description		iDecoderObjectID long 1 In Decoder object ID
lOldObjID long 1 In Old decoder object ID	20	piModelint * 1-65535 2 Out Pointer to decoder
iNewAddr int 2 In New decoder address		type ID
plNewObjID long * 1 Out New decoder object ID		1 Opaque object ID handle returned by
Opaque object ID handle returned by KamDecoderPutAdd.		KamDecoderPutAdd. 2 Maximum value for this server given by
2 1–127 for short locomotive addresses. 1–10239 for		KamDecoderGetMaxModels.
long locomotive decoders. 0-511 for accessory decoders.	25	
Return Value Type Range Description		iError short 1 Error flag
Error short 1 Error flag		iError = 0 for success. Nonzero is an error number
iError = 0 for success. Nonzero is an error number		(see KamMiscGetErrorMsg).
(see KamMiscGetErrorMsg). KamDecoderChangeOldNewAddr takes an old decoder object ID		KamDecoderGetModelFromObj takes a decoder object ID and pointer to a decoder type ID as parameters. It sets the
and a new decoder address as parameters. It moves the	30	memory pointed to by piModel to the decoder type ID
specified locomotive or accessory decoder to iNewAddr and		associated with iDCCAddr.
sets the memory pointed to by plNewObjID to the new		0KamDecoderGetModelFacility
object ID. The old object ID is now invalid and should		Parameter List Type Range Direction Description
no longer be used.		IDecoderObjectID long 1 In Decoder object ID
0KamDecoderMovePort Parameter List Type Range Direction Description	25	pdwFacility long * 2 Out Pointer to decoder facility mask
DecoderObjectID long 1 In Decoder object ID	35	1 Opaque object ID handle returned by
iLogicalPortID int 1-65535 2 In Logical port ID		KamDecoderPutAdd.
1 Opaque object ID handle returned by		2 0 - DCODE_PRGMODE_ADDR
KamDecoderPutAdd.		1 - DCODE_PRGMODE_REG
2 Maximum value for this server given by KamPortGetMaxLogPorts.		2 - DCODE_PRGMODE_PAGE 3 - DCODE_PRGMODE_DIR
Return Value Type Range Description	40	4 - DCODE_PRGMODE_FLYSHT
iError short 1 Error flag		5 - DCODE_PRGMODE_FLYLNG
1 iError = 0 for success. Nonzero is an error number		6 - Reserved
(see KamMiscGetErrorMsg).		7 - Reserved 8 - Reserved
KamDecoderMovePort takes a decoder object ID and logical port ID as parameters. It moves the decoder specified by		9 - Reserved
DecoderObjectID to the controller specified by	45	
iLogicalPortID.		11 - Reserved
0KamDecoderGetPort		12 - Reserved
Parameter List Type Range Direction Description IDecoderObjectID long 1 In Decoder object ID		13 - DCODE_FEAT_DIRLIGHT
lDecoderObjectID long 1 In Decoder object ID piLogicalPortID int * 1-65535 2 Out Pointer to		14 - DCODE_FEAT_LNGADDR 15 - DCODE_FEAT_CVENABLE
logical port ID	50	
1 Opaque object ID handle returned by		17 - DCODE_FEDMODE_REG
KamDecoderPutAdd.		18 - DCODE_FEDMODE_PAGE
2 Maximum value for this server given by KamPortGetMaxLogPorts.		19 - DCODE_FEDMODE_DIR 20 - DCODE_FEDMODE_FLYSHT
Return Value Type Range Description		21 - DCODE_FEDMODE_FLYLNG
iError short 1 Error flag	55	Return Value Type Range Description
1 iError = 0 for success. Nonzero is an error number		iError short 1 Error flag
(see KamMiscGetErrorMsg).		iError = 0 for success. Nonzero is an error number
KamDecoderMovePort takes a decoder object ID and pointer		(see KamMiscGetErrorMsg). KamDecoderGetModelFacility takes a decoder object ID and
to a logical port ID as parameters. It sets the memory pointed to by piLogicalPortID to the logical port ID		pointer to a decoder facility mask as parameters. It
associated with IDecoderObjectID.		sets the memory pointed to by pdwFacility to the decoder
0KamDecoderCheckAddrInUse	60	facility mask associated with iDCCAddr.
Parameter List Type Range Direction Description		0KamDecoderGetObjCount
iDecoderAddress int 1 In Decoder address		Parameter List Type Range Direction Description
iLogicalPortID int 2 In Logical Port ID iDecoderClass int 3 In Class of decoder		iDecoderClass int 1 In Class of decoder piObjCount int * 0-65535 Out Count of active
1 Opaque object ID handle returned by		decoders
KamDecoderPutAdd.	65	1 1 - DECODER_ENGINE_TYPE,
2 Maximum value for this server given by		2 - DECODER_SWITCH_TYPE,

0 - retain state, 1 - clear state.

Range

iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).
KamDecoderPutDel takes a decoder object ID and clear flag

as parameters. It deletes the locomotive object specified

by IDecoderObjectID from the locomotive database.

Error flag

Description*

Type

Return Value

0KamDecoderGetMfgName

iError short

US 6,530,329 B2

23 -continued

24 -continued

1 Opaque object ID handle returned by KamDecoderPutAdd.

emergency stop for all modes

FALSE.

Return Value

Speed range is dependent on whether the decoder is

Forward is boolean TRUE and reverse is boolean

Description

set to 14, 18, or 128 speed steps and matches the values defined by NMRA S9.2 and RP 9.2.1. 0 is stop and 1 is

APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE 3 - DECODER_SENSOR_TYPE. Parameter List Range Direction Description Туре IDecoderObjectID long Range Decoder object ID Return Value Туре Description* BSTR * 2 Error flag pbsMfgName Out Pointer to iError short iError = 0 for success. Nonzero is an error number manufacturer name Opaque object ID handle returned by (see KamMiscGetErrorMsg). KamDecoderGetObjCount takes a decoder class and a pointer KamDecoderPutAdd. to an address count as parameters. It sets the memory Exact return type depends on language. It is Cstring * for C++. Empty string on error. pointed to by piObjCount to the count of active decoders of the type given by iDecoderClass. 0KamDecoderGetObjAtIndex Return Value Туре iError short Error flag iError = 0 for success. Nonzero is an error number Parameter List Туре Range Direction Description (see KamMiscGetErrorMsg). In Decoder array index iIndex int iDecoderClass Class of decoder KamDecoderGetMfgName takes a decoder object ID and In plDecoderObjectID long * pointer to a manufacturer name string as parameters. It On Pointer to decoder sets the memory pointed to by pbsMfgName to the name of object ID the decoder manufacturer. 0KamDecoderGetPowerMode 0 to (KamDecoderGetAddressCount - 1). 2 1 - DECODER ENGINE TYPE. Range Description 2 - DECODER_SWITCH_TYPE, Parameter List Турс Direction IDecoderObjectID long pbsPowerMode BSTR * Decoder object TD 3 - DECODER_SENSOR_TYPE. Ιn Out Pointer to Opaque object ID handle returned by decoder power KamDecoderPutAdd. Return Value Description mode Туре Opaque object ID handle returned by Error flag iError short KamDecoderPutAdd. iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamDecoderGetObjCount takes a decoder index, decoder Exact return type depends on language. It is Cstring * for C++. Empty string on error. class, and a pointer to an object ID as parameters. It Return Value Туре Range Description^e Error flag sets the memory pointed to by plDecoderObjectID to the iError short iError = 0 for success. Nonzero is an error number selected object ID. (see KamMiscGetErrorMsg). 0KamDecoderPutAdd KamDecoderGetPowerMode takes a decoder object ID and a Parameter List Type Range Direction Description pointer to the power mode string as parameters. It sets iDecoderAddress Decoder address 1-65535 2 the memory pointed to by pbsPowerMode to the decoder iLogicalCmdPortID int Logical In power mode. 0KamDecoderGetMaxSpeed command port ID 1-65535 2 Parameter List Direction Description iLogicalProgPortID int In Logical Type Range programming port ID lDecoderObjectID long In Decoder object ID Out Pointer to max piSpeedStep Clear state flag iClearState int 3 speed step 1 Opaque object ID handle returned by KamDecoderPutAdd. iModel int In Decoder model type ID plDecoderObjectID long * 5 Out Decoder 14, 28, 56, or 128 for locomotive decoders. 0 for object ID accessory decoders. Return Value 1-127 for short locomotive addresses. 1-10239 for long locomotive decoders. 0-511 for accessory decoders. Туре Description Range Maximum value for this server given by iError short Error flag KamPortGetMaxLogPorts. 3 0 - retain state, 1 - clear state. iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). Maximum value for this server given by KamDecoderGetMaxSpeed takes a decoder object ID and a pointer to the maximum supported speed step as parameters. It sets the memory pointed to by piSpeedStep KamDecoderGetMaxModels. Opaque object ID handle. The object ID is used to A. Commands to control locomotive decoders This section describes the commands that reference the decoder. Range Return Value Туре Description Error flag iError short iError = 0 for success. Nonzero is an error number control locomotive decoders. These commands control (see KamMiscGetErrorMsg). things such as locomotive speed and direction. For things such as tocomotive speed and direction. For efficiency, a copy of all the engine variables such speed is stored in the server. Commands such as KamEngGetSpeed communicate only with the server, not the actual decoder. You should first make any changes to the server copy of KamDecoderPutAdd takes a decoder object ID, command logical port, programming logical port, clear flag, decoder model ID, and a pointer to a decoder object ID as parameters. It creates a new locomotive object in the the engine variables. You can send all changes to the engine using the KamCmdCommand command. 0KamEngGetSpeed locomotive database and sets the memory pointed to by plDecoderObjectID to the decoder object ID used by the server as a key 0KamDecoderPutDel Parameter List Type Range Direction Description lDecoderObjectID Decoder object ID Type Range Direction long Parameter List Description Ιn 1DecoderObjectID long 1 Decoder object ID lpSpeed Out Pointer to locomotive 2 Ι'n Clear state flag speed iClearState int * 3 1 Opaque object ID handle returned by KamDecoderPutAdd. **InDirection** Out Pointer to locomotive

-continued

26

-continued

APPLICATION PROGRAMMING INTERFACE

APPLICATION PROGRAMMING INTERFACE

iError short	1	Erro	or flag		5	2 FL is 0. F1-	F8 are 1	-8 respect	ively. Maxir	num for
1 iError = 0 fo			o is an error	number		this decoder is give				
(see KamMiscGetErrorMsg). KamEngGetSpeed takes the decoder object ID and pointers						Function active is b FALSE.	oolean T	RUE and	inactive is t	ooolean
to locations to store							Туре	Ran	ge	Description
as parameters. It set							1		or flag	- · · · · · · · · · · · ·
to the locomotive sp					10				o is an error	number
lpDirection to the lo	∞moti	ve directio	n.			(see KamMiscGetE			11	. C
0KamEngPutSpeed		ъ	D'	D. Luthur		KamEngGetFunctio				
Parameter List		Range	Direction In	Description• Decoder object ID		ID, and a pointer to function state as pa				
IDecoderObjectID iSpeed	long int *	1 2	In	Locomotive speed		to by lpFunction to				Omica
iDirection	int *	3	In	Locomotive direction	15	0KamEngPutFuncti				
1 Opaque obje					13	Parameter List		Range	Direction	Description
KamDecoderPutAdo			•			lDecoderObjectID	long	1	In	Decoder object ID
2 Speed range						iFunctionID	int	0-82	In	Function ID number
set to 14, 18, or 128						iFunction	int	3 adla satus	In	Function value
defined by NMRA			U is stop ar	10. 1 18		 Opaque objet KamDecoderPutAd 		illule tetui	neu by	
emergency stop for 3 Forward is b			reverse is b	ooolean	20	2 FL is 0. F1-		-8 respec	tively. Maxi	mum for
FALSE.						this decoder is give				
	Туре	Ran	ige	Description		3 Function act				
	1		or flag			boolean FALSE.	_			
1 iError = 0 fo			o is an error	number			Туре	Ran		Description•
(see KamMiscGetE			11 470		25		1		or flag	
KamEngPutSpeed to					23	1 iError = 0 for (see KamMiscGetE			o is an error	namoei
locomotive speed, a parameters. It sets t						KamEngPutFunctio			r object ID.	a function
iSpeed and the loco						ID, and a new func				
iDirection. Note: Th						specified locomotiv				
locomotive database	e. The	data is not	sent to the d	ecoder		iFunction. Note: Th				
until execution of the					30	locomotive databas				
set to the maximum			lecoder if iS	peed		until execution of t		CmdComn	nand comma	nd.
exceeds the decoder		3. .				OKamEngGetFunct Parameter List		Range	Direction	Description
OKamEngGetSpeed Parameter List		Dance	Direction	Description		lDecoderObjectID	long	1	In	Decoder object ID
lDecoderObjectID		Range 1	In	Decoder object ID		piMaxFunction	int *	0−8	Out	Pointer to maximum
	int *	14,28,128	Out	Pointer to number	35	primar univers		•		function number
-population-p		,,		of speed steps	33	1 Opaque obj	ect ID ha	andle retur	ned by	
 Opaque objet 	ect ID l	nandle retui	nea by			KamDecoderPutAd	ld.			
KamDecoderPutAd	d.		•			Return Value	Туре	Ran		Description
KamDecoderPutAde Return Value	d. Type	Rai	1ge	Description		Return Value iError short	Type 1	Err	or flag	•
KamDecoderPutAde Return Value iError short	d. Type 1	Rai Err	nge or flag	-		Return Value iError short 1 iError = 0 f	Type 1 or succes	Erress. Nonzei		•
KamDecoderPutAde Return Value iError short 1 iError = 0 for	d. Type 1 or succ	Rai Err ess. Nonzei	nge or flag	-	40	Return Value iError short 1 iError = 0 f (see KamMiscGetE	Type 1 or succes rrorMsg	Erress. Nonzei).	or flag o is an error	r number
KamDecoderPutAde Return Value iError short 1 iError = 0 for (see KamMiscGetE	d. Type 1 or succ rrorMs	Ran Erress. Nonzen g).	nge or flag o is an error	r number	40	Return Value iError short 1 iError = 0 f	Type 1 or succes frorMsg onMax ta	Erross. Nonzei). ikes a deo	or flag o is an error oder object l	r number ID and a
KamDecoderPutAde Return Value iError short 1 iError = 0 for	d. Type 1 or succ rrorMs steps tal	Ran Erress. Nonzer g). kes the dec	nge or flag o is an error	r number	40	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maxi sets the memory po	Type 1 or succes crorMsg onMax ta imum fur ointed to	Erross. Nonzer). kes a deconction ID : by piMax	or flag to is an error oder object l as parameter Function to	r number ID and a rs. It the
KamDecoderPutAdı Return Value iError short 1 iError = 0 fc (see KamMiscGete KamEngGetSpeedS pointer to a location as a parameter. It se	d. Type 1 or succ rrorMs teps ta n to sto ets the	Ran Erress. Nonzer g). kes the dec re the num memory po	nge or flag oo is an error oder object l ber of speed ointed to by	r number	40	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunction pointer to the maxisets the memory points maximum possible	Type 1 or succes crorMsg onMax ta imum fur ointed to	Erross. Nonzer). kes a deconction ID : by piMax	or flag to is an error oder object l as parameter Function to	r number ID and a rs. It the
KamDecoderPutAdı Return Value iError short i iError = 0 ft (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It sa lpSpeedSteps to the	d. Type 1 or succe frorMs steps tal n to sto ets the number	Ran Erress. Nonzer g). kes the dec re the num memory po	nge or flag oo is an error oder object l ber of speed ointed to by	r number	40	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunction pointer to the maxisets the memory pointer maximum possible decoder.	Type 1 for successive	Erross. Nonzer). kes a deconction ID : by piMax	or flag to is an error oder object l as parameter Function to	r number ID and a rs. It the
KamDecoderPutAde Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st lpSpeedSteps to the 0KamEngPutSpeed:	d. Type 1 or succerrorMs steps tal n to sto ets the numbe Steps	Ran Erress. Nonzer g). kes the dec re the num memory poer of speed	or flag to is an error oder object l ber of speed ointed to by steps.	r number ID and a I steps		Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maxisets the memory promaximum possible decoder. 0KamEngGetName	Type 1 for success frorMsg onMax ta imum fur ointed to function	Erross. Nonzer). ikes a deo nction ID : by piMax number f	or flag to is an error oder object I as parameter Function to for the specif	r number ID and a rs. It the
KamDecoderPutAde Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List	d. Type 1 or succerrorMs Steps tal n to sto ets the numbe Steps Type	Range Range Range Range Range	or flag or flag or is an error oder object l ber of speed ointed to by steps. Direction	r number ID and a steps Description	40	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFuncti pointer to the maxisets the memory p maximum possible decoder. 0KamEngGetName Parameter List	Type 1 for success frorMsg, onMax ta imum fur ointed to function Type	Erross. Nonzer). ikes a deo nction ID : by piMax number f Range	or flag to is an error oder object l as parameter Function to or the specif	r number ID and a s. It the fied Description
KamDecoderPutAd Return Value iError short 1 iError = 0 ft (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID	d. Type 1 or succerrorMs Steps tal n to sto ets the numbe Steps Type long	Ran Erress. Nonzer g). kes the dec re the num memory poer er of speed Range	oge or flag or is an error oder object l ber of speed ointed to by steps. Direction In	r number ID and a steps Description Decoder object ID		Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctio pointer to the maxi- sets the memory po maximum possible decoder. 0KamEngGetName Parameter List lDecoderObjectID	Type 1 for success frorMsg, onMax ta imum fur ointed to function Type	Erross. Nonzer). kes a deconction ID: by piMax number f Range	or flag to is an error oder object I as parameter Function to for the specif	r number ID and a rs. It the
KamDecoderPutAd Return Value iError short 1 iError = 0 ft (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID	d. Type 1 or succerrorMs Steps tal n to sto ets the numbe Steps Type	Range Range Range Range Range	oge or flag or is an error oder object l ber of speed ointed to by steps. Direction In	r number ID and a steps Description		Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFuncti pointer to the maxisets the memory p maximum possible decoder. 0KamEngGetName Parameter List	Type 1 for success frorMsg, onMax ta imum fur ointed to function Type long	Erross. Nonzer). kes a deconction ID: by piMax number f Range	or flag to is an error oder object I as parameter Function to or the specif Direction In	r number ID and a s. It the fied Description Decoder object ID
KamDecoderPutAd Return Value iError short 1 iError = 0 ft (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID	d. Type 1 or succerrorMs Steps tal n to sto ests the e numb Steps Type long int	Range 1 14,28,128	or flag or flag or is an error oder object l ber of speed ointed to by steps. Direction In	r number ID and a steps Description Decoder object ID Locomotive speed		Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctio pointer to the maxi- sets the memory po maximum possible decoder. 0KamEngGetName Parameter List lDecoderObjectID	Type 1 for success recording to success recording t	Erross. Nonzer). ikes a deo oction ID : by piMax number f Range 1	or flag to is an error oder object I as parameter Function to for the specific Direction In Out	r number ID and a s. It the fied Description Decoder object ID Pointer to
KamDecoderPutAda Return Value iError short 1 iError = 0 ft (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd	d. Type 1 or succerrorMs steps talento stoets the enumber Steps Type long int	Range 1 14,28,128	or flag or flag or is an error oder object l ber of speed ointed to by steps. Direction In	number ID and a steps Description Decoder object ID Locomotive speed steps	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctiv pointer to the maxi- sets the memory po- maximum possible decoder. 0KamEngGetName Parameter List 1DecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAde	Type 1 or succese FrorMsg, on Max taken fur on the function function Type long BSTR * sect ID held.	Erross. Nonzer characteristics a deconction ID a by piMax number f Range 1 2 andle return	or flag o is an error oder object l as parameter Function to or the specif Direction In Out	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name
KamDecoderPutAde Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value	d. Type 1 or succerrorMs steps talen to sto en numbe Steps Type long int ect ID d. Type	Range 1 14,28,128 handle returned.	onge or flag to is an error oder object l ber of speed ointed to by steps. Direction In In rned by	r number ID and a steps Description Decoder object ID Locomotive speed		Return Value iError short iError = 0 f (see KamMiscGetE KamEngGetFuncti pointer to the maxi sets the memory p maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return	Type 1 for success	Erross. Nonzer ss. Nonzer ukes a deo cotion ID by piMax number f Range 1 2 andle retur	or flag to is an error toder object l as parameter frunction to toor the specif Direction In Out rned by language. It	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name
KamDecoderPutAda Return Value iError short i iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed. Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short	d. Type 1 or succe fror Ms fteps tal n to sto ets the numb Steps Type long int ect ID d. Type 1	Range 1 14,28,128 handle return Range Errors Range Ran	onge or flag or is an error oder object l ber of speed inted to by steps. Direction In In raned by onge or flag	r number ID and a steps Description Decoder object ID Locomotive speed steps Description	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maximum possible decoder. 0KamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C+++.	Type 1 for success	Erross. Nonzer skes a deo cotton ID by piMax number f Range 1 2 andle retur pends on etring on e	or flag o is an error oder object I as parameter Frunction to for the specifi Direction In Out rned by language. It rror.	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name
KamDecoderPutAda Return Value iError short i iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short i iError = 0 fc	d. Type 1 or succe free hard to store the enumber of the store that the enumber of the free hard to store the enumber of the free hard to store the enumber of the free hard to store t	Range 1 14,28,128 handle returess. Nonzergess. Nonzergess the decrease of speed Range 1 14,28,128 handle returess. Nonzergess. Nonzergess.	onge or flag or is an error oder object l ber of speed inted to by steps. Direction In In raned by onge or flag	r number ID and a steps Description Decoder object ID Locomotive speed steps Description	45	Return Value iError = 0 f (see KamMiscGetE KamEngGetFunctio pointer to the maxi- sets the memory po- maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1	Type 1 for success rorms g, on Max taimum fur ointed to function Type long BSTR * lect ID hadd. In type de Empty s Type	Erross. Nonzer). likes a deo action ID by piMax number f Range 1 2 andle return pends on string on e	or flag o is an error oder object las parameter Frunction to for the special Direction In Out rned by language. It rror, age	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name
KamDecoderPutAde Return Value iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngTutSpeedF Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE	d. Type 1 Type 1 to r succe strorMs steps tal n to sto ets the e numb Steps Type long int d. Type 1 or succe strorMs	Range 1 14,28,128 handle returness. Nonzer	onge or flag to is an error oder object l ber of speed inted to by steps. Direction In freed by or flag or flag or is an error	r number ID and a steps Description Decoder object ID Locomotive speed steps Description T number	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory pomaximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short	Type 1 1 for success rivers Msg. on Max taken mum fur ointed to function BSTR • 1 feet ID hadd. In type de Empty s Type 1	Erross. Nonzer). ikes a deo nction ID o by piMax number f Range 1 2 andle retur pends on tring on e Ran Err	or flag o is an error oder object I as parameter Frunction to for the specifi Direction In Out rened by language. It rror, age or flag	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name is Description
KamDecoderPutAda Return Value iError short i iError = 0 fc (see KamMissGetE KamEngGetSpeedS pointer to a location as a parameter. It se lpSpeedSteps to the 0KamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short i iError = 0 fc (see KamMissGetE KamEngPutSpeedS	d. Type 1 Type 1 or succerrorMs steps tal n to sto ets the enumb Steps Type long int tect ID d d Type 1 or succerrorMs steps tal for succerrorMs	Range 1 14,28,128 handle returness. Nonzer gg).	onge or flag or is an error oder object leber of speed inted to by steps. Direction In	r number ID and a steps Description Decoder object ID Locomotive speed steps Description r number ID and a new	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory pomaximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short	Type 1 1 or success irrorMsg, on Max to imum fur object to function BSTR • Lect ID hadd, in type de Empty s Type 1 1 for succes	Erross. Nonzer). ikes a deo oction ID o by piMax number f Range 1 2 andle retur epends on string on e Ran Erross. Nonzer	or flag o is an error oder object las parameter Frunction to for the special Direction In Out rned by language. It rror, age	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name is Description
KamDecoderPutAde Return Value iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngTutSpeedF Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE	d. Type 1 Type 1 to r succeirrorMs teps tain to sto ets the enumb Steps Type long int ect ID d. Type 1 or succeirrorMs teps tain	Range 1 14,28,128 handle returness. Nonzer gg).	onge or flag or is an error oder object leber of speed inted to by steps. Direction In	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maxis sets the memory pomaximum possible decoder. 0KamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f	Type 1 1 or success rorms g, on Max taken imum fur obinted to function 2 Type long BSTR * lect ID hadd. In type de Empty s Type 1 for success rorms g	Erress. Nonzer). okes a deo oction ID oby piMax number f Range 1 2 andle return pends on string on e Range Services. Nonzer Services. Nonzer).	or flag o is an error oder object las parameter Frunction to for the special Direction In Out rned by language. It rror, age or flag ro is an erro	r number ID and a s.s. It the field Description Decoder object ID Pointer to locomotive name is Description
KamDecoderPutAde Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed stop short Note: This comman	d. Type 1 or succe cirrorMs steps tal n to sto te to the n unmb Steps Type long int Type 1 or succerrorMs teps tal eps as	Range 1 14,28,128 handle returness. Nonzer gg). Range 2 1 14,28,128 handle returness. Nonzer gg). Range the deca parameter house the deca changes the data changes the changes the data changes the data changes the deca parameter house data changes the deca changes the decaparameter house data changes data changes the decaparameter house data changes data chang	onge or flag or is an error oder object lober of speed inted to by steps. Direction In	Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database.	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maxisets the memory py maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive name the commotive name the commotive name control points of the commotive name the commotive name control points of the commotive name control points of the commotive name commotive n	Type 1 1 or successfror Msg. on Max takes a consultation of function of functi	Erross. Nonzer b. Nonzer ches a deo cotion ID o by pi Max number f Range 1 2 andle retur pends on etring on e Ran Erross. Nonzer b. decoder oh rameters. 1	or flag o is an error oder object Is as parameter Frunction to or the specifi Direction In Out rend by language. It rror, age or flag to is an error oject ID and it sets the me	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory
KamDecoderPutAda Return Value iError short i iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It se IpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed steps in th Note: This comman The data is not sen	d. Type 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Rangers. Nonzergy. Rangers of speed Range 1 14,28,128 handle returness. Nonzergy. Rangers of speed Range 1 changes the decoder under the data coloning the decoder under the	onge or flag or is an error oder object leber of speed inted to by steps. Direction In	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the maximum possible decoder. 0KamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring for C++. Return Value iError short 1 iError = 0 f (see KamMiscGet KamEngGetName to locomotive nam pointed to by pbsE	Type 1 1 for successfrorMsg, on Max taken interest to function a Type long BSTR * lect ID hadd. In type de Empty s Type 1 for successfrorMsg takes a c me as paringName a sparingName	Erross. Nonzer b. Nonzer ches a deo cotion ID o by pi Max number f Range 1 2 andle retur pends on etring on e Ran Erross. Nonzer b. decoder oh rameters. 1	or flag o is an error oder object Is as parameter Frunction to or the specifi Direction In Out rend by language. It rror, age or flag to is an error oject ID and it sets the me	r number ID and a s. It the fied Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory
KamDecoderPutAde Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComm	d. Type 1 1 or succerrorMs iteps to the content of	Range 1 14,28,128 handle returness. Nonzer Range 1 14,28,128 handle returness. Nonzer ges. Nonzer ges. Nonzer the decoder unmand. K	or flag or of flag or of speed or of flag or of flag or of sa an error or oder object I or is an error or oder object I or is the speed or of speed or	Description Decoder object ID Locomotive speed steps Description Trumber ID and a new number edSteps. e database. n of SetMaxSpeed returns	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive to the maxisets the memory of maximum possible decoder. 0KamEngGetName Parameter List 1DecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nam pointed to by pbsE 0KamEngPutName both the commotive nam pointed to by pbsE 0KamEngPutName the locomotive nam	Type 1 1 or successfrorMsg, on Max ta timum fur to inted to function BSTR * eet ID hadd. In type de Empty s Type 1 for successfrorMsg takes a cene as par EngName EngName	Erross. Nonzer). class a deo cation ID o by piMax number f Range 1 2 andle retur pends on etring on e Ran Err ss. Nonzer). decoder oh cameters. I to the nat	or flag o is an error oder object las parameter Frunction to for the specifi Direction In Out rned by language. It rror. age or flag ro is an erro oject ID and it sets the m me of the lo	r number ID and a s.s. It the field Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive.
KamDecoderPutAda Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi	d. Type 1 1 1 or successfrom Ms. Steps tal. It is the tall. It is the	Range 1 14,28,128 handle returness. Nonzer gl. kes the decre the num memory per of speed Range 1 14,28,128 handle returness. Nonzer gl. kes the deca a parameter notive data changes the decoder u ommand. Keed for the	onge or flag or is an error oder object lober of speed inted to by steps. Direction In	Description Decoder object ID Locomotive speed steps Description Trumber ID and a new number edSteps. e database. n of SelMaxSpeed returns error is	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory py maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nan pointed to by pbsE OKamEngPutName Parameter List	Type 1 1 or successfror Msg. on Max ta timum fur jointed to of function as Type long BSTR * (set ID hadden type de Empty s Type 1 for successfror Msg. takes a 6 one as paringName is Type	Erross. Nonzer b. Nonzer ches a deo oction ID oby pi Max number f Range 1 2 andle return pends on ottring on e Range France San Nonzer checoder ob rameters. I to the nat	or flag o is an error oder object I as parameter Frunction to for the specifi Direction In Out rend by language. It rror. age or flag ro is an error oject ID and it sets the me me of the lo Direction	Description Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description
KamDecoderPutAda Return Value iError short i iError = 0 fc (see KamMissGetE KamEngGetSpeedS pointer to a location as a parameter. It se IpSpeedSteps to the OKamEngPutSpeed Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iiError = 0 fc (see KamMissGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atter	d. Type 1 1 1 or successfrom Ms. Steps tal. It is the tall. It is the	Range 1 14,28,128 handle returness. Nonzer gl. kes the decre the num memory per of speed Range 1 14,28,128 handle returness. Nonzer gl. kes the deca a parameter notive data changes the decoder u ommand. Keed for the	onge or flag or is an error oder object lober of speed inted to by steps. Direction In	Description Decoder object ID Locomotive speed steps Description Trumber ID and a new number edSteps. e database. n of SelMaxSpeed returns error is	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctic pointer to the max sets the memory promaximum possible decoder. 0KamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName be locomotive nar pointed to by pbsE 0KamEngPutName parameter List IDecoderObjectID	Type 1 1 or successfrorMsg, on Max taken intended to function a Type long BSTR * lect ID hadd. In type de Empty s Type 1 for successfrorMsg takes a contained to general taken a contained taken a c	Erross. Nonzer Andrews a deo Andrews a deo Andrews a deo Andrews a deo By piMax Anumber f Range 1 2 Andle retur Andle retur Andrews andrews Berross. Nonzer Anumeters. 1 to the na Range 1	or flag o is an error oder object I as parameter Fruction to for the special Direction In Out rened by language. It rror. age or flag ro is an erro oject ID and it sets the m me of the lo Direction In	r number ID and a s.s. It the field Description Decoder object ID Pointer to locomotive name is Description I number a pointer to emory comotive. Description Descri
KamDecoderPutAda Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi	d. Type 1 1 ror succeptor and the sumble succeptor and the succeptor and t	Range 1 14,28,128 handle returness. Nonzer gl. kes the decre the num memory per of speed Range 1 14,28,128 handle returness. Nonzer gl. kes the deca a parameter notive data changes the decoder u ommand. Keed for the	onge or flag or is an error oder object lober of speed inted to by steps. Direction In	Description Decoder object ID Locomotive speed steps Description Trumber ID and a new number edSteps. e database. n of SelMaxSpeed returns error is	45	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory py maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nan pointed to by pbsE OKamEngPutName Parameter List	Type 1 1 or successfrorMsg, on Max taken to function function to f	Erross. Nonzer). lekes a deo oction ID is by piMax number f Range 1 2 2 andle return pends on string on e Range Erross. Nonzer). lecoder obtameters. I to the nate Range 1 2 2	or flag o is an error oder object I as parameter Frunction to for the specifi Direction In Out rened by language. It rror, age or flag ro is an erro rject ID and it sets the m me of the lo Direction In Out	Description Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description
KamDecoderPutAda Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the 00KamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed sto of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. 0KamEngGetFunct Parameter List	d. Type 1 1 respectively. Type 1 n to stoets the enumber of the stoets the enumber of the stoets the first of the stoets the enumber of the stoets of the st	Range 1 14,28,128 handle returness. Nonzer gl. kes the decre the num memory per of speed Range 1 14,28,128 handle returness. Nonzer gl. kes the deca a parameter notive data changes the decoder u ommand. Keed for the	onge or flag or is an error oder object lober of speed inted to by steps. Direction In	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of letMaxSpeed returns error is steps Description	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxisets the memory py maximum possible decoder. 0KamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nan pointed to by pbsE 0KamEngPutName Parameter List IDecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd FamengPutName Parameter List IDecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd FamengPutName 1 Opaque obj KamDecoderPutName 1 Opaque 0	Type 1 1 or successfror Msg. on Max taken as parengo Manuel Manue	Erross. Nonzer Annual Properties Range	or flag o is an error oder object I as parameter Frunction to for the special Direction In Out red by language. It rror. age or flag to is an error oject ID and it sets the me me of the lo Direction In Out rred by	Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name
KamDecoderPutAda Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed st of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. 0KamEngGetFunct Parameter List IDecoderObjectID	d. Type 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Range 1 14,28,128 handle returness. Nonzer g). Range 1 14,28,128 handle returness. Nonzer g). Res the deco a parameter decoder u ommand. Keed for the made to see Range 1	onge or flag or is an error oder object l ber of speed inted to by steps. Direction In	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of GetMaxSpeed returns error is steps Description Decoder object ID	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxisets the memory pomaximum possible decoder. OKamEngGetName Parameter List 1DecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nam pointed to by pbsE OKamEngPutName Parameter List 1DecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parareter Standard Company of the point of the	Type 1 1 or successfrorMsg, on Max ta timum fur to inted to function at the successfrorMsg BSTR * lect ID hadd. In type de Empty s Type 1 for successfrorMsg takes a cone as paraing Name at the successfrorMsg barbara for successfrorMsg takes a cone as paraing Name at the successfrorMsg takes a cone as paraing Name at the successfrorMsg takes a cone as paraing Name at the successfrorMsg takes a cone as paraing Name at the successfrorMsg takes a cone as paraing Name at the successfrorMsg takes a cone at	Erross. Nonzer Annual Properties Range	or flag o is an error oder object I as parameter Frunction to for the specifi Direction In Out rened by language. It rror, age or flag ro is an erro rject ID and it sets the m me of the lo Direction In Out	Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name
KamDecoderPutAde Return Value iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngTutSpeedC Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed sto of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. OKamEngGetFunct Parameter List IDecoderObjectID iFunctionID	d. Type 1 1 respectively. Type	Range 1 14,28,128 handle returness. Nonzer gl. Range 1 14,28,128 handle returness. Nonzer gl. Range 1 to the first proper of speed handle returness. Nonzer gl. Range st. And the decoder unmand. Keed for the made to see Range gl. 1 0–8 2	onge or flag or is an error oder object l ber of speed inted to by steps. Direction In In rened by or flag or is an error oder object l comodive the base to iSpe to locomotive the object l comotive	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of ietMaxSpeed returns error is steps Description Decoder object ID Function ID number	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory pointer to the memory pointer the memo	Type 1 1 or successfor Msg. on Max ta timum fur to function of function of the	Erross. Nonzer See Nonzer See Nonzer See Nonzer See Nonzer Range 1 2 andle retur spends on string on e Ran Erross. Nonzer See Nonzer to the nat Range 1 2 andle retur spends on string on e Ran Erross. Nonzer Lecoder ob rameters. I to the nat Range 1 2 andle retur spends on etring on e Ran Erross. Nonzer Lecoder ob rameters. I	or flag o is an error oder object las parameter Frunction to or the specifi Direction In Out rned by language. It rror, rage or flag ro is an error oject ID and it sets the m me of the lo Direction In Out rned by on language on flag on or flag or of the lo or flag or of the lo	Description Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name
KamDecoderPutAda Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. OKamEngGetFunct Parameter List IDecoderObjectID iFunctionID IpFunction	d. Type 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Range 1 14,28,128 handle returness. Nonzer gl. Range 1 14,28,128 handle returness. Nonzer gl. Range 1 to the first proper of speed handle returness. Nonzer gl. Range st. And the decoder unmand. Keed for the made to see Range gl. 1 0–8 2	onge or flag or is an error oder object l ber of speed inted to by steps. Direction In	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of GetMaxSpeed returns error is steps Description Decoder object ID	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive pointer to the maxis sets the memory py maximum possible decoder. OKamEngGetName Parameter List IDecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError short 2 iExact parate List IDecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parate LPCSTR for C++. Return Value	Type 1 1 for successfrorMsg, on Max taken to function	Erross. Nonzer Range Range Randle retur pends on etring on e Ran Erross. Nonzer Range Range Randle retur pends on etring on e Ran Erross. Nonzer Range	or flag o is an error oder object Is as parameter Frunction to or the special Direction In Out red by language. It rror, age or flag to is an error oject ID and it sets the me me of the lo Direction In Out out red by on language. It of the lo on language. Out on language.	Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name
KamDecoderPutAda Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It st IpSpeedSteps to the 0KamEngPutSpeed Parameter List IDecoderObjectID iSpeedSteps 1 Opaque obje KamDecoderPutAd Return Value iError short 1 iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. 0KamEngGetFunct Parameter List IDecoderObjectID iFunctionID IpFunction value	d. Type 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Range 1 14,28,128 handle return changes the decoretion of the decoder unmand. Keed for the made to see a garantee to see	or flag or of speed inted to by steps. Direction In In In sets the base to iSpee I sets the base to iSpee I condition am DecoderO decoder. An the speed is Direction In In In Out	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of ietMaxSpeed returns error is steps Description Decoder object ID Function ID number	45 50	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive from the maximum possible decoder. OKamEngGetName Parameter List 1DecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nam pointed to by pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater List 1DecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater List 1DecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater LPCSTR for C++. Return Value iError short	Type 1 1 or successfrorMsg, on Max ta timum fur to inted to function at the successfrorMsg BSTR * lect ID hadd. In type de Empty s Type 1 for successfrorMsg takes a cone as paringName at the successfrorMsg BSTR to me as paringName at the successfrorMsg takes a cone as paringName at the successfromMsg takes a cone at the successfrom the successf	Erress. Nonzer). okes a deo oction ID o by piMax number f Range 1 2 andle retur pends on string on e Ran Erress. Nonzer). lecoder ob rameters. I to the na Range 1 2 andle retur edepends Range 1 Rang	or flag o is an error oder object I as parameter Function to for the special Direction In Out rend by language. It rror, age or flag ror is an erro riget ID and it sets the me it sets the me of the lo Direction In Out rned by on language or flag on of language or flag on flag on of language or flag	Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name
KamDecoderPutAda Return Value iError short iError = 0 fc (see KamMiscGetE KamEngGetSpeedS pointer to a location as a parameter. It selpSpeedSteps to the OKamEngPutSpeed: Parameter List IDecoderObjectID iSpeedSteps Opaque obje KamDecoderPutAd Return Value iError short iError = 0 fc (see KamMiscGetE KamEngPutSpeedS number of speed st of speed steps in th Note: This comman The data is not sen the KamCmdComn the maximum possi generated if an atte beyond this value. OKamEngGetFunct Parameter List IDecoderObjectID iFunctionID IpFunction	d. Type 1 1 or succertrorMs teps tal n to sto ets the en umb Steps Type long int Type 1 to the eect ID to d. Type 1 to the ible spe mpt is ion Typ long int int "to the controrMs to the to t	Range 1 14,28,128 handle return changes the decoretion of the decoder unmand. Keed for the made to see a garantee to see	or flag or of speed inted to by steps. Direction In In In sets the base to iSpee I sets the base to iSpee I condition am DecoderO decoder. An the speed is Direction In In In Out	Description Description Decoder object ID Locomotive speed steps Description r number ID and a new number edSteps. e database. n of ietMaxSpeed returns error is steps Description Decoder object ID Function ID number	45 50 55	Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetFunctive from the maximum possible decoder. OKamEngGetName Parameter List 1DecoderObjectID pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact return Cstring * for C++. Return Value iError short 1 iError = 0 f (see KamMiscGetE KamEngGetName the locomotive nam pointed to by pbsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater List 1DecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater List 1DecoderObjectID bsEngName 1 Opaque obj KamDecoderPutAd 2 Exact parater LPCSTR for C++. Return Value iError short	Type 1 1 or successfror Msg. on Max ta timum fur to inted to of function and the successfrom t	Erress. Nonzer See See See See See See See See See Se	or flag o is an error oder object Is as parameter Frunction to or the special Direction In Out red by language. It rror, age or flag to is an error oject ID and it sets the me me of the lo Direction In Out out red by on language. It of the lo on language. Out on language.	Description Description Decoder object ID Pointer to locomotive name is Description r number a pointer to emory comotive. Description Decoder object ID Locomotive name

27

-continued

28

APPLICATION PROGRAMMING INTERFACE KamEngPutName takes a decoder object ID and a BSTR as parameters. It sets the symbolic locomotive name to bsEngName 0KamEngGetFunctionName Parameter List Туре Range Direction Description 1DecoderObjectID Decoder object ID long In Function ID number In iFunctionID int BSTR * 3 Out Pointer to pbsFcnNameString function name Opaque object ID handle returned by KamDecoderPutAdd. 2 FL is 0. F1-F8 are 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax. 3 Exact return type depends on language. It is Cstring * for C++. Empty string on error. Return Value Type Range iError short Error flag iError* = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamEngGetFuncationName takes a decoder object ID, function ID, and a pointer to the function name as parameters. It sets the memory pointed to by pbsFcnNameString to the symbolic name of the specified function. 0KamEngPutFunctionName Parameter List 1yp-IDecoderObjectID long int Type Range Direction Description Decoder object ID In 0-82 In Function ID number bsFcnNameString BSTR 3 In Function name 1 Opaque object ID handle returned by KamDecoderPutAdd. FL is 0. F1-F8 are 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax. Exact parameter type depends on language. It is LPCSTR for C++. Туре Range Description Return Value Error Flag iError short iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamEngPutFunctionName takes a decoder object ID, function ID, and a BSTR as parameters. It sets the specified symbolic function name to bsFcnNameString. 0KamEngGetConsistMax Parameter List Type Range Direction Description IDecoderObjectID long 1 piMaxConsist int * 2 Decoder object ID In Pointer to max consist number Opaque object ID handle returned by KamDecoderPutAdd. Command station dependent. Return Value Туре Description Range iError short Error flag iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamEngGetConsistMax takes the decoder object ID and a pointer to a location to store the maximum consist as parameters. It sets the location pointed to by piMaxConsist to the maximum number of locomotives that can but placed in a command station controlled consist. Note that this command is designed for command station consisting. CV consisting is handled using the CV commands. 0KamEngPutConsistParent Range Parameter List Туре Direction Description lDCCParentObjID long 1 In Parent decoder iDCCAliasAddr 2 Alias decoder address Opaque object ID handle returned by KamDecoderPutAdd. 1-127 for short locomotive addresses. 1-10239 for long locomotive decoders. Return Value Type Range Description

Error flag iError = 0 for success. Nonzero is an error number

KamEngPutConsistParent takes the parent object ID and an

alias address as parameters. It makes the decoder

iError short

(see KamMiscGetErrorMsg).

-continued

		-continued								
-		APPLICATION PROGRAMMING INTERFACE								
	5	specified by IDCCParentObjID the consist parent referred to by iDCCAliasAddr. Note that this command is designed for command station consisting. CV consisting is handled								
	10	using the CV commands. If a new parent is defined for a consist; the old parent becomes a child in the consist. To delete a parent in a consist without deleting the consist, you must add a new parent then delete the old								
		parent using KamEng 0KamEngPutConsiste Parameter List	Child Type	Range	Direction	Description				
		IDCCParentObjID IDCCObjID	long	1	In In	Parent decoder object ID Decoder object ID				
	15	 Opaque object KamDecoderPutAdd 	t ID ha	ndle retu	rned by					
		Return Value T iError short 1 1 iError = 0 for		Er	nge for flag ro is an erro	Description or number				
	20	(see KamMiscGetErr KamEngPutConsistC and decoder object I	orMsg) hild tak	esthede	coder parer	nt object ID				
		decoder specified by by IDCCParentObjII for command station	IDCCC D. Note consist	bjID to that this ing. CV	the consist in command is consisting is	identified s designed s handled				
	25	using the CV comma the parent has not be KamEngPutConsistP	en set p arent.	previousl		s invalid if				
	20	0KamEngPutConsist Parameter List lDecoderObjectID 1 Opaque object KamDecoderPutAdd	Type long at ID ha	Range 1	Direction In Irned by	Description Decoder object ID				
	30		уре		nge ror flag	Description				
		1 iError = 0 fo (see KamMiscGetEr. KamEngPutConsistF	rorMsg)	ı						
	35		ves the om the	decoder consist.	specified by Note that th	is				
		consisting is handled the parent is remove A. Commands to	l using d, all cl	the CV o	commands. It re removed	Note: If also.				
	4 0	This sec control accessory de	tion des coders.	cribes th These co	e command ommands co	s that introl				
		things such as acces efficiency, a copy of is stored in the serve KamAccGetFunction	all the er. Com	engine v mands st	ariables suc ich as	h speed				
	45	the actual decoder. In the server copy of the all changes to the er- command.	ne engir	e variab	les. You can	send				
		0KamAccGetFunction Parameter List	on Type	Range	Direction	Description				
	50	lDecoderObjectID iFunctionID lpFunction	long int int *	1 0-31 2 3	In In Out	Decoder object ID Function ID number Pointer to function value				
		1 Opaque obje KamDecoderPutAdd 2 Maximum fo	l.							
	55	KamAccGetFunction 3 Function acti boolean FALSE.		oolean T	RUE and in	active is				
		Return Value	Гуре 1	E	ange ror flag	Description				
	60	(see KamMiscGetEn KamAccGetFunction ID, and a pointer to	rorMsg n takes the loc). the deco ation to	der object II store the spe	D, a function ecified				
		function state as part to by lpFunction to 0KamAccGetFuncti	the spec	cified fur	nction state.					
	65	Parameter List IDecoderObjectID piValue	Type long int *	Range 1 2	Direction In Out	Description Decoder object ID Function bit mask				

30

iError = 0 for success. Nonzero is an error number

-continued -continued APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE Opaque object ID handle returned by KamDecoderPutAdd KamDecoderPutAdd Exact return type depends on language. It is Cstring * for C++. Empty string on error. Each bit represents a single function state. Range Maximum for this decoder is given by Return Value Type Description KamAccGetFunctionMax. iError short 1 Error flag iError = 0 for success. Nonzero is an error number Return Value Туре Description Range iError short Error flag (see KamMiscGetErrorMsg). KamAccGetName takes a decoder object ID and a pointer to iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). a string as parameters. It sets the memory pointed to by KamAccGetFunctionAll takes the decoder object ID and a pbsAccNameString to the name of the accessory. 0KamAccPutName pointer to a bit mask as parameters. It sets each bit in the memory pointed to by piValue to the corresponding Parameter List Range Direction Type Description long BSTR lDecoderObjectID In Decoder object ID bsAccNameString 0KamAccPutFunction In Accessory name 1 Opaque object ID handle returned by KamDecoderPutAdd. Parameter List Type Range Direction Description 1DecoderObjectID Decoder object ID long Exact parameter type depends on language. It is iFunctionID int 0-31 2 In Function ID number iFunction Function value LPCSTR for C++. Opaque object ID handle returned by Return Value Туре Range Description KamDecoderPutAdd. iError short Error flag Maximum for this decoder is given by iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamAccGetFunctionMax. Function active is boolean TRUE and inactive is KamAccPutName takes a decoder object ID and a BSTR as boolean FALSE. parameters. It sets the symbolic accessory name to bs AccName Return Value Туре Range Error flag 0KamAccGetFunctionName iError short iError = 0 for success. Nonzero is an error number Parameter List Range Direction Туре Description (see KamMiscGetErrorMsg). lDecoderObjectID long Decoder object ID KamAccPutFunction takes the decoder object ID, a function iFunctionID int 0 - 312In Function ID number pbsFcnNameString BSTR * 3 ID, and a new function state as parameters. It sets the Pointer to specified accessory database function state to lFunction. function name 1 Opaque object ID handle returned by KamDecoderPutAdd. Note: This command only changes the accessory database. The data is not sent to the decoder until execution of the KamCmdCommand command. Maximum for this decoder is given by 0KamAccPutFunctionAll KamAccGetFunctionMax. Parameter List Type Range Direction Description Exact return type depends on language. It is IDecoderObjectID long Decoder object ID Cstring * for C++. Empty string on error. Return Value Type Range Ĭ'n Pointer to function state Туре Description* iError short Error flag array 1 Opaque object ID handle returned by KamDecoderPutAdd. iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamAccGetFuncationName takes a decoder object ID, Each bit represents a single function state. function ID, and a pointer to a string as parameters. It sets the memory pointed to by pbsFcnNameString to the symbolic name of the specified function. Maximum for this decoder is given by KamAccGetFunctionMax. Return Value Туре Description Range Error flag iError short 0KamAccPutFunctionName iError = 0 for success. Nonzero is an error number Parameter List Туре Range Direction Description (see KamMiscGetErrorMsg). 1DecoderObjectID long Decoder object ID KamAccPutFunctionAll takes the decoder object ID and a iFunction ID int 0-31 2 In Function ID number BSTR bit mask as parameters. It sets all decoder function bsFcnNameString Function enable states to match the state bits in iValue. The 45 Opaque object ID handle returned by possible enable states are TRUE and FALSE. The data is not sent to the decoder until execution of the KamDecoderPutAdd. Maximum for this decoder is given by KamCmdCommand command KamAccGetFunctionMax. 3 Exact parameter type depends on language. It is LPCSTR for C++. 0KamAccGetFunctionMax Parameter List Range Direction Туре Description 1DecoderObjectID long Decoder object ID Return Value Туре Range piMaxFunction Error flag 0-31.2 Out Pointer to maximum iError short iError = 0 for success. Nonzero is an error number function number (see KamMiscGetErrorMsg). Opaque object ID handle returned by KamDecoderPutAdd. KamAccPutFunctionName takes a decoder object ID, function Maximum for this decoder is given by ID, and a BSTR as parameters. It sets the specified KamAccGetFunctionMax. symbolic function name to bsFcnNameString. Range Error flag 0KamAccRegFeedback Parameter List Type Return Value Type Description Direction iError short Type Range Description* iError = 0 for success. Nonzero is an error number lDecoderObjectID long Decoder object ID In (see KamMiscGetErrorMsg). KamAccGetFunctionMax takes a decoder object ID and bsAccNode BSTR In Server node name iFunctionID 0-31 3 In Function ID number int pointer to the maximum function number as parameters. It Opaque object ID handle returned by sets the memory pointed to by piMaxFunction to the maximum possible function number for the specified KamDecoderPutAdd. Exact parameter type depends on language. It is LPCSTR for C++. decoder. 0KamAccGetName Maximum for this decoder is given by Parameter List Туре Range Direction Description KamAccGetFunctionMax. lDecoderObjectID long 1 BSTP * 2 Decoder object ID Return Value Туре Description pbsAccNameString BSTP * 2 Ou 1 Opaque object ID handle returned by Error flag Out Accessory name iError short

31

32

-continued

-continued

APPLICATION PROGRAMMING INTERFACE		APPLICATION PROGRAMMING INTERFACE
(see KamMiscGetErrorMsg).	5	such as controlling command station power. The steps to
KamAccRegFeedback takes a decoder object ID, node name		control a given command station vary depending on the
string, and function ID, as parameters. It registers		type of command station. 0KamOprPutTurnOnStation
interest in the function given by iFunctionID by the method given by the node name string bsAccNode.		Parameter List Type Range Direction Description
bsAccNode identifies the server application and method to		iLogicalPortID int 1-65535 1 In Logical port ID
call if the function changes state. Its format is	10	
"\\{Server\\{App\}. {Method}" where {Server} is the server		KamPortGetMaxLogPorts. Return Value Type Range Description
name, {App} is the application name, and {Method} is the method name.		iError short 1 Error flag
0KamAccRegFeedbackAll		1 iError = 0 for success. Nonzero is an error number
Parameter List Type Range Direction Description		(see KamMiscGetErrorMsg).
IDecoderObjectID long 1	15	KamOprPutTurnOnStation takes a logical port ID as a parameter. It performs the steps necessary to turn on
bsAccNode BSTR 2 In Server node name 1 Opaque object ID handle returned by		the command station. This command performs a combination
KamDecoderPutAdd.		of other commands such as KamOprPutStartStation,
2 Exact parameter type depends on language. It is		KamOprPutClearStation, and KamOprPutPowerOn.
LPCSTR for C++. Return Value Type Range Description		0KamOprPutStartStation Parameter List Type Range Direction Description
iError short 1 Error flag	20	iLogicalPortID int 1-65535 1 In Logical port ID
1 iError = 0 for success. Nonzero is an error number		1 Maximum value for this server given by
(see KamMiscGetErrorMsg).		KamPortGetMaxLogPorts.
KamAccRegFeedbackAll takes a decoder object ID and node name string as parameters. It registers interest in all		Return Value Type Range Description iError short 1 Error flag
functions by the method given by the node name string		1 iError = 0 for success. Nonzero is an error number
bsAccNode. bsAccNode identifies the server application	25	(500 1201111110001101111000)
and method to call if the function changes state. Its		KamOprPutStartStation takes a logical port ID as a parameter. It performs the steps necessary to start the
format is "\\{Server}\{App}. {Method}" where {Server} is the server name, {App} is the application name, and		command station.
{Method} is the method name.		0KamOprPutClearStation
0KamAccDelFeedback		Parameter List Type Range Direction Description
Parameter List Type Range Direction Description IDecoderObjectID long 1 In Decoder object ID	30	iLogicalPortID int 1-65535 1 In Logical port ID Maximum value for this server given by
bsAccNode BSTR 2 In Server node name		KamPortGetMaxLogPorts.
iFunctionID int 0-31 3 In Function ID number		Return Value Type Range Description
1 Opaque object ID handle returned by		iError short 1 Error flag
KamDecoderPutAdd. Exact parameter type depends on language. It is	25	1 iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).
LPCSTR for C++.	33	KamOprPutClearStation takes a logical port ID as a
3 Maximum for this decoder is given by		parameter. It performs the steps necessary to clear the
KamAccGetFunctionMax.		command station queue.
Return Value Type Range Description iError short 1 Error flag		0KamOprPutStopStation Parameter List Type Range Direction Description
iError = 0 for success. Nonzero is an error number	40	iLogicalPortID int 1-65535 1 In Logical port ID
(see KamMiscGetErrorMsg).	40	1 Maximum value for this server given by
KamAccDelFeedback takes a decoder object ID, node name string, and function ID, as parameters. It deletes		KamPortGetMaxLogPorts. Return Value Type Range Description
interest in the function given by iFunctionID by the		iError short 1 Error flag
method given by the node name string bsAccNode.		1 iError = 0 for success. Nonzero is an error number
bsAccNode identifies the server application and method to	45	(see KamMiscGetErrorMsg). KamOprPutStopStation takes a logical port ID as a
call if the function changes state. Its format is "\\Server\\{App\}.{Method}" where {Server} is the server		parameter. It performs the steps necessary to stop the
name, {App} is the application name, and {Method} is the		command station.
method name.		0KamOprPutPowerOn
OKamAccDelFeedbackAll Parameter List Type Range Direction Description•		Parameter List Type Range Direction Description iLogicalPortID int 1-65535 1 In Logical port ID
Parameter List Type Range Direction Description IDecoder Object ID long 1 In Decoder object ID	50	1 Maximum value for this server given by
bsAccNode BSTR 2 In Server node name		KamPortGetMaxLogPorts.
1 Opaque object ID handle returned by		Return Value Type Range Description
KamDecoderPutAdd. Exact parameter type depends on language. It is		iError short 1 Error flag 1 iError = 0 for success. Nonzero is an error number
LPCSTR for C++.		(see KamMiscGetErrorMsg).
Return Value Type Range Description	55	KamOprPutPowerOn takes a logical port ID as a parameter.
iError short 1 Error flag		It performs the steps necessary to apply power to the
1 iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).		track. 0KamOprPutPowerOff
KamAccDelFeedbackAll takes a decoder object ID and node		Parameter List Type Range Direction Description
name string as parameters. It deletes interest in all		iLogicalPortID int 1-65535 1 In Logical port ID
functions by the method given by the node name string	60	1 Maximum value for this server given by KamPortGetMaxLogPorts.
bsAccNode. bsAccNode identifies the server application and method to call if the function changes state. Its		Return Value Type Range Description
format is "\\{Server}\{App}.{Method}" where {Server} is		iError short 1 Error flag
the server name, {App} is the application name, and		1 iError = 0 for success. Nonzero is an error number
{Method} is the method name.A. Commands to control the command station		(see KamMiscGetErrorMsg). KamOprPutPowerOff takes a logical port ID as a parameter.
This section describes the commands that	65	
control the command station. These commands do things		track.

33

RATE 0 - 300 BAUD, 1 - 1200 BAUD, 2 - 2400 BAUD,

3 - 4800 BAUD, 4 - 9600 BAUD, 5 - 14400 BAUD,

34

-continued -continued APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE 6 - 16400 BAUD, 7 - 19200 BAUD 0KamOprPutHardReset Туре Range Di 1-65535 1 In Direction Description 2 PARITYO - NONE, 1 - ODD, 2 - EVEN, 3 - MARK, iLogicalPortID 4 - SPACE int Logical port ID Maximum value tor this server given by 0 - 1 bit, 1 - 1.5 bits, 2 - 2 bits KamPortGetMaxLogPorts. WATCHDOG 500 - 65535 milliseconds. Recommended value 2048 Description Return Value Type Range FLOW 0 - NONE, 1 - XON/XOFF, 2 - RTS/CTS, 3 BOTH DATA 0 - 7 bits, 1 - 8 bits iError short Error flag iError = 0 for success. Nonzero is an error number DEBUGBit mask. Bit 1 sends messages to debug file. (see KamMiscGetErrorMsg). Bit 2 sends messages to the screen. Bit 3 shows queue data. Bit 4 shows UI status. Bit 5 is KamOprPutHardReset takes a logical port ID as a parameter. It performs the steps necessary to perform a reserved. Bit 6 shows semaphore and critical hard reset of the command station. sections. Bit 7 shows miscellaneous messages. Bit 0KamOprPutEmergencyStop Type Range Di int 1-65535 1 In 8 shows comm port activity. 130 decimal is recommended for debugging. Parameter List Description Direction iLogical Port IDLogical port ID Maximum value for this server given by PARALLEL KamPortGetMaxLogPorts. 0KamPortPutConfig Parameter List Type Range Direction Description^e Return Value Туре Range Description iLogicalPortID int iError short Error flag 1-65535 1 In Logical port ID 20 iError. = 0 for success. Nonzero is an error number iIndex int In Configuration type index iValue In Configuration value (see KamMiscGetErrorMsg). int In KamOprPutEmergencyStop takes a logical port ID as a iKey int Debug key Maximum value for this server given by parameter. It performs the steps necessary to broadcast an emergency stop command to all decoders. KamPortGetMaxLogPorts. 0KamOprGetStationStatus See FIG. 7: Controller configuration Index values Range Parameter List Type Direction Description for a table of indexes and values 1-65535 1 In Logical port ID Used only for the DEBUG iIndex value. Should be set iLogicalPortID int BSTR * 2 Command station to 0. Return Value status string Туре Range Description Maximum value for this server given by iError short Error flag KamPortGetMaxLogPorts. iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). Exact return type depends on language. It is KamPortPutConfig takes a logical port ID, configuration Cstring * for C++. Return Value index, configuration value, and key as parameters. It sets the port parameter specified by iIndex to the value specified by iValue. For the DEBUG iIndex value, the Range Туре Description Error flag iError short iError = 0 for success. Nonzero is an error number debug file path is C:\Temp\Debug{PORT}.txt where {PORT} (see KamMiscGetErrorMsg). is the physical comm port ID. 0KamPortGetConfig KamOprGetStationStatus takes a logical port ID and a pointer to a string as parameters. It set the memory pointed to by pbsCmdStat to the command station status. The exact format of the status BSTR is vendor dependent. Parameter List Type iLogicalPortID int Range Dis 1--65535 1 In Direction Description Logical port ID Configuration type index Commands to configure the command station int pi Value int * 2 Out Pointer to configuration value communication port Maximum value for this server given by This section describes the commands that KamPortGetMaxLogPorts. configure the command station communication port. These 2 See FIG. 7: Controller configuration Index values for a table of indexes and values. commands do things such as setting BAUD rate. Several of the commands in this section use the numeric controller ID (iControllerID) to identify a specific type of Return Value Туре Range Description command station controller. The following table shows the mapping between the controller ID (iControllerID) and iError short Error flag iError = 0 for success. Nonzero is an error number controller name (bsControllerName) for a given type of (see KamMiscGetErrorMsg). command station controller. KamPortGetConfig takes a logical port ID, configuration iControllerID bsControllerName index, and a pointer to a configuration value as parameters. It sets the memory pointed to by piValue to UNKNOWN Unknown controller type 0 SIMULAT Interface simulator the specified configuration value. LENZ_1x LENZ_2x Lenz version 1 serial support module Lenz version 2 serial support module 0KamPortGetName Parameter List Type Direction Description Range DIGIT_DT200 Digitrax direct drive support using iPhysicalPortID int 1-65535 1 In Physical port DT200 number pbsPortName BSTR * 2 DIGIT_DCS100 Digitrax direct drive support using Physical port name DCS100 Maximum value for this server given by KamPortGetMaxPhysical. MASTERSERIES North coast engineering master 6 Exact return type depends on language. It is series Cstring * for C++. Empty string on error. Return Value Type Range System one RAMFIxx system SYSTEMONE Туре RAMFIX Range Description SERIAL NMRA serial interface Error flag iError short iError = 0 for success. Nonzero is an error number 10 EASYDCC MPK6050 CVP Easy DCC Marklin 6050 interface (AC and DC) (see KamMiscGetErrorMsg). 11 12 MPK6023 Marklin 6023 interface (AC) KamPortGetName takes a physical port ID number and a pointer to a port name string as parameters. It sets the memory pointed to by pbsPortName to the physical port name such as "COMM1." Digitrax direct drive using PR1 Direct drive interface routine 13 DIGIT_PR1 DIRECT 14 15 ZTC system ltd 16 TRIX TRIX controller 0KamPortPutMapController iIndex iValue Values Parameter List Type Range Direction Description Name RETRANS 10-255 iLogicalPortID 1-65535 1 In Logical port ID

iControllerID

int

1-65535 2 In

Command station

type ID

It disconnects the server to the specified command

station.

US 6,530,329 B2

35

36

-continued -continued APPLICATION PROGRAMMING INTERFACE APPLICATION PROGRAMMING INTERFACE iCommPortID 1-65535 3 In Physical comm 0KamCmdCommand Parameter List port ID Type Range Direction Description IDecoderObjectID derObjectID long 1 In Opaque object ID handle returned by Decoder object ID Maximum value for this server given by KamPortGetMaxLogPorts. See FIG. 6: Controller ID to controller name KamDecoderPutAdd. mapping for values. Maximum value for this server is Return Value Туре Description Range given by KamMiscMaxControllerID. Error flag iError short iError = 0 for success. Nonzero is an error number Maximum value tor this server given by KamPortGetMaxPhysical. (see KamMiscGetErrorMsg). Return Value Range Description KamCmdCommand takes the decoder object ID as a parameter. Туре iError short Error flag It sends all state changes from the server database to the specified locomotive or accessory decoder. iError = 0 for success. Nonzero is an error number Cab Control Commands (see KamMiscGetErrorMsg). KamPortPutMapController takes a logical port ID, a This section describes commands that control the cabs attached to a command station. command station type ID, and a physical communications port ID as parameters. It maps iLogicalPortID to 0KamCabGetMessage Parameter List Type iCommPortID for the type of command station specified by Range Direction Description 1-65535 1 In Cab address iCabAddress int iControllerID. 0KamPortGetMaxLogPorts BSTR * 2 Out Cab message string Parameter List Type Range piMaxLogicalPorts int * 1 Direction Description• Maximum value is command station dependent. Exact return type depends on language. It is Out Maximum logical Cstring * for C++. Empty string on error. Return Value Type Range port ID Normally 1-65535. 0 returned on error Description Error flag Range iError short Return Value Туре Description iError = 0 for success. Nonzero is an error number Error flag iError short iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamCabGetMessage takes a cab address and a pointer to a (see KamMiscGetErrorMsg). KamPortGetMaxLogPorts takes a pointer to a logical port message string as parameters. It sets the memory pointed to by pbsMsg to the present cab message. 0KamCabPutMessage ID as a parameter. It sets the memory pointed to by piMaxLogicalPorts to the maximum logical port ID. OKamPortGetMaxPhysical Parameter List Range Direction Description Туре Direction Parameter List Type int * Range Description iCabAddress int In Cab address BSTR 2 bs Msg Out Cab message string pMaxPhysical 1 Out Maximum physical port ID Maximum value is command station dependent. 2 Exact parameter type depends on language. It is LPCSTR for C++. pMaxSerial int * 1 Out Maximum serial port ID Maximum parallel Return Value Туре int * 1 Range Out pMaxParallel Error flag port ID iError short Normally 1-65535. 0 returned on error. iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). Туре Range Description Return Value iError short Error flag KamCabPutMessage takes a cab address and a BSTR as iError = 0 for success. Nonzero is an error number parameters. It sets the cab message to bsMsg. 0KamCabGetCabAddr (see KamMiscGetErrorMsg). Parameter List 1712 IncoderObjectID long 1 In int * 1-65535 2 Out Direction KamPortGetMaxPhysical takes a pointer to the number of Description* Decoder object ID physical ports, the number of serial ports, and the number of parallel ports as parameters. It sets the Pointer to Cab memory pointed to by the parameters to the associated address Opaque object ID handle returned by values Commands that control command flow to the command KamDecoderPutAdd. A. Maximum value is command station dependent. station Туре Descriptioni This section describes the commands that Return Value Range iError short Error flag control the command flow to the command station. These iError = 0 for success. Nonzero is an error number commands do things such as connecting and disconnecting (see KamMiscGetErrorMsg). from the command station. 0KamCmdConnect KamCabGetCabAddr takes a decoder object ID and a pointer Type Range Dir int 1-65535 1 In to a cab address as parameters. It set the memory pointed to by piCabAddress to the address of the cab Parameter List Direction Description• iLogicalPortID Logical port ID Maximum value for this server given by attached to the specified decoder. 0KamCabPutAddrToCab KamPortGetMaxLogPorts. Return Value Range Description Parameter List Type Range Direction Description Туре DecoderObjectID long 1 Error flag In Decoder object ID iError short 1-65535 2 In iError = 0 for success. Nonzero is an error number iCabAddress int Cab address Opaque object ID handle returned by (see KamMiscGetErrorMsg). KamCmdConnect takes a logical port ID as a parameter. It KamDecoderPutAdd. Maximum value is command station dependent connects the server to the specified command station. Return Value Туре Range Description 0KamCmdDisConnect Type Range Dir int 1-65535 1 In iError short Error flag Parameter List Direction Description iError = 0 for success. Nonzero is an error number iLogicalPortID Logical port ID Maximum value for this server given by (see KamMiscGetErrorMsg). KamPortGetMaxLogPorts. KamCabPutAddrToCab takes a decoder object ID and cab address as parameters. It attaches the decoder specified Return Value Type Range Description by iDCCAddr to the cab specified by iCabAddress. Error flag iError short iError = 0 for success. Nonzero is an error number Miscellaneous Commands This section describes miscellaneous commands (see KamMiscGetErrorMsg). KamCmdDisConnect takes a logical port ID as a parameter. that do not fit into the other categories.

0KamMiscGetErrorMsg Parameter List Type Range

Direction Description

38

-continued

US 6,530,329 B2

37

-continued

APPLICATION PROGRAMMING INTERFACE			INTERFACE		APPLICATION PROGRAMMING INTERFACE	
iError	int	0-65535 1	[n	Error flag	- 5	automatically whenever the server stops running. Demo
			ro indicates a			versions of the program cannot save data and this command
Return Value	Туре	Ras		Description		will return an error in that case.
bsErrorString	BSTR		_	Error string		0KamMiscGetControllerName
			language. It	is		Parameter List Type Range Direction Description
Cstring for C++. I						iControllerID int 1-65535 1 In Command station
KamMiscGetError					10	74
It returns a BSTR				•		pbsName BSTR * 2 Out Command station type
message associate 0KamMiscGetClo			error mag.			name See FIG. 6: Controller ID to controller name
Parameter List	Type	Range	Direction	Description		1 See FIG. 6: Controller ID to controller name mapping for values. Maximum value for this server is
iLogicalPortID	int	1-65535 1		Logical port ID		given by KamMiscMaxControllerID.
iSelectTimeMode		2	In	Clock source	15	
piDay	int *	0-6	Out	Day of week	15	Cstring * for C++. Empty string on error.
piHours	int *	0-23	Out	Hours		Return Value Type Range Description
piMinutes	int *	0-59	Out	Minutes		pbsName BSTR 1 Command station type name
piRatio	int *	3	Out	Fast clock ratio		Return Value Type Range Description
		or this serve	r given by			iError short 1 Error flag
KamPortGetMaxL			_		20	1 iError = 0 for success. Nonzero is an error number
			on and sync			(see KamMiscGetErrorMsg).
1 - Load direct fre			from cache	i server		KamMiscGetControllerName takes a command station type ID
copy of command						and a pointer to a type name string as parameters. It
3 Real time Return Value				Description		sets the memory pointed to by pbsName to the command
iError short	Type 1	Ran	or flag	Description		station type name. 0KamMiscGetControllerNameAtPort
			o is an error	number	25	
(see KamMiscGet			C III UII OITO	namovi		iLogicalPortID int 1-65535 1 In Logical port ID
KamMiscGetCloc			t ID, the tim	ne mode, and		pbsName BSTR * 2 Out Command station type
pointers to locatio						name
and fast clock rati						1 Maximum value for this server given by
pointed to by piD	ay to th	e fast clock	day, sets po	inted		KamPortGetMaxLogPorts.
to by piHours to t	he fast	clock hours,	sets the me	mory	30	2 Exact return type depends on language. It is
pointed to by piM						Cstring * for C++. Empty string on error.
the memory point						Return Value Type Range Description
The servers local			d if the com	mand .		iError short 1 Error flag
station does not su		fast clock.				iError = 0 for success. Nonzero is an error number
0KamMiscPutClo		D	Direction	Description		(see KamMiscGetErrorMsg).
Parameter List	int	Range 1-65535 1		Description	35	KamMiscGetControllerName takes a logical port ID and a
iLogicalPortID iDay	int	0-6	in In	Logical port ID Day of week		pointer to a command station type name as parameters. It
iHours	int	0-23	ln	Hours		sets the memory pointed to by pbsName to the command station type name for that logical port.
iMinutes	int	0-59	In	Minutes		OKamMiscGetCommandStationValue
iRatio	int	2	In	Fast clock ratio		Parameter List Type Range Direction Description
1 Maximum	value f	or this serve	r given by		40	iControllerID int 1-65535 1 In Command station
KamPortGetMax1	ogPorts	. 2 Real tim	e clock ratio),	40	type ID
Return Value	Type	Ran		Description		iLogicalPortID int 1-65535 2 In Logical port ID
iError short	1		or flag			iIndex int 3 In Command station array index
			o is an error	number		pi Value int * 0-65535 Out Command station value
(see KamMiscGet			11. 1	-1		See FIG. 6: Controller ID to controller name
KamMiscPutClock					45	mapping for values. Maximum value for this server is
the fast clock day, minutes, and the f					,,,	given by KamMiscMaxControllerID. Maximum value for this server given by
the fast clock usin				SCIS		KamPortGetMaxLogPorts.
0KamMiscGetInte			.			3 0 to KamMiscGetCommandStationIndex .
Parameter List	Тур		Direction	n Description		Return Value Type Range Description
pbsInterfaceVersion			Out	Pointer to interface		iError short 1 Error flag
-				version string	50	1 iError = 0 for success. Nonzero is an error number
1 Exact return	n type	depends on l	language. It	is		(see KamMiscGetErrorMsg).
Cstring * for C++						KamMiscGetCommandStationValue takes the controller ID,
Return Value	Type	Ran		Description		logical port, value array index, and a pointer to the
iError short	1		or flag			location to store the selected value. It sets the memory
			o is an error	number		pointed to by piValue to the specified command station
(see KamMiscGet KamMiscGetInter			naintar ta a		55	miscellaneous data value. OKamMiscSetCommandStationValue
interface version s				1		
memory pointed to				interface		Parameter List Type Range Direction Description iControllerID int 1-65535 1 In Command station
version string. The						type ID
lines depending or						iLogicalPortID int 1-65535 2 In Logical port ID
0KamMiscSaveDa			••			illuder int 3 In Command station agent index
Parameter List	Тур	e Range	Direction	Description	60	iValue int 0-65535 In Command station Value
NONE						See FIG. 6: Controller ID to controller name
Return Value	Type	Ran		Description		mapping for values. Maximum value for this server is
iError short	1		or flag			given by KamMiscMaxControllorID.
			o is an error	number		2 Maximum value for this server given by
(see KamMiscGet			re It cares	all server	65	KamPortGetMaxLogPorts. 3 0 to
KamMiscSaveDate data to permanent				an scivel	03	
and to be munch	rotage	. Lum CUMIII	TOTAL 19 INII			Return Value Type Range Description

39

-continued

40

APPLICATION PROGRAMMING INTERFACE

iError short	1	Erro	or flag	
1 iError = 0	for suc	cess. Nonzer-	o is an erro	r number
(see KamMiscGe	tErrorM	sg).		
KamMiscSetCon	nmandSt	ation Value ta	kes the con	troller ID.
logical port, valu				
data value. It set				
to the value give				
0KamMiscGetCo				
Parameter List		Range	Direction	Description
iControllerID	IIIt	1-65535 1	ın	Command station type ID
iControllerID iLogicalPortID	int	1-65535 2		

See FIG. 6: Controller ID to controller name mapping for values. Maximum value for this server is given by KamMiscMaxControllerID.

Maximum value for this server given by

KamPortGetMaxLogPorts.Return Value Туре

Description Range iError short Error flag

iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).

KamMiscGetCommandStationIndex takes the controller ID, logical port, and a pointer to the location to store the maximum index. It sets the memory pointed to by pilndex to the specified command station maximum miscellaneous data index.

0KamMiscMaxControllerID

Parameter List Type Range Dire piMaxControllerID int * 1-65535 1 Out Direction Description Maximum controller type ID

See FIG. 6: Controller ID to controller name mapping for a list of controller ID values. 0 returned on error.

Return Value Type Description Range Error flag iError short

iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).

KamMiscMaxControllerID takes a pointer to the maximum controller ID as a parameter. It sets the memory pointed to by piMaxControllerID to the maximum controller type

0KamMiscGetControllerFacility Parameter List Type Range

iControllerID 1-65535 1 In Command station type ID pdwFacility long * 2 Pointer to command station facility mask

Direction

Description

See FIG. 6: Controller ID to controller name

mapping for values. Maximum value for this server is given by KamMiscMaxControllerID.

- 0 CMDSDTA_PRGMODE ADDR 1 CMDSDTA_PRGMODE_REG
- 2 CMDSDTA_PRGMODE_PAGE
- 3 CMDSDTA_PRGMODE_DIR 4 CMDSDTA_PRGMODE_FLYSHT
- 5 CMDSDTA_PRGMODE_FLYLNG
- 6 Reserved
- 7 Reserved
- 8 Reserved
- 9 Reserved
- 10 CMDSDTA_SUPPORT_CONSIST
- 11 CMDSDTA_SUPPORT_LONG 12 CMDSDTA_SUPPORT_FEED
- 13 CMDSDTA_SUPPORT_2TRK
- 14 CMDSDTA_PROGRAM_TRACK 15 CMDSDTA_PROGMAM_POFF
- 16 CMDSDTA_FEDMODE_ADDR
- 17 CMDSDTA FEDMODE REG - CMDSDTA_FEDMODE_PAGE
- 19 CMDSDTA_FEDMODE_DIR 20 CMDSDTA_FEDMODE_FLYSHT
- 21 CMDSDTA_FEDMODE_FLYLNG
- 30 Reserved

31 - CMDSDTA_SUPPORT_FASTCLK

Return Value Туре Description

-continued APPLICATION PROGRAMMING INTERFACE

iError short Error flag iError = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamMiscGetControllerFacility takes the controller ID and a pointer to the location to store the selected controller facility mask. It sets the memory pointed to

by pdwFacility to the specified command station facility

The terms and expressions which have been employed in the foregoing specification are used therein as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims which follow.

What is claimed is:

- 1. A method of operating a digitally controlled model railroad comprising the steps of:
- (a) transmitting a first command from a first program to an interface:
- (b) transmitting a second command from a second program to said interface; and
- (c) sending third and fourth commands from said interface representative of said first and second commands, respectively, to a digital command station.
- 2. The method of claim 1, further comprising the steps of:
- (a) providing an acknowledgment to said first program in response to receiving said first command by said interface prior to sending said third command to said digital command station; and
- (b) providing an acknowledgment to said second program in response to receiving said second command by said interface prior to sending said fourth command to said digital command station.
- 3. The method of claim 2, further comprising the steps of:
- (a) selectively sending said third command to one of a plurality of digital command stations; and
- (b) selectively sending said fourth command to one of said plurality of digital command stations.
- 4. The method of claim 3, further comprising the step of 45 receiving command station responses representative of the state of said digitally controlled model railroad from said plurality of digital command stations.
 - 5. The method of claim 4, further comprising the step of comparing said command station responses to previous commands sent to at least one of said plurality of digital command stations to determine which of said previous commands it corresponds with.
 - 6. The method of claim 5, further comprising the steps of:
 - (a) maintaining a sending queue of commands to be transmitted to said plurality of digital command stations; and
 - (b) retransmitting at least one of said commands in said sending queue periodically until removed from said sending queue as a result of the comparison of said command station responses to previous commands.
- 7. The method of claim 6, further comprising the step of updating a database of the state of said digitally controlled model railroad based upon said receiving command station responses representative of said state of said digitally con-65 trolled model railroad.
 - 8. The method of claim 7, further comprising the step of providing said acknowledgment to said first program in

response to receiving said first command by said interface together with state information from said database related to said first command.

- 9. The method of claim 8 wherein said first command and said third command are the same command, and said second command and said fourth command are the same command.
- 10. A method of operating a digitally controlled model railroad comprising the steps of:
 - (a) transmitting a first command from a first program to an interface; and
 - (b) said interface selectively sending a second command representative of said first command to one of a plurality of digital command stations based upon information contained within at least one of said first and second commands.
- 11. The method of claim 10, further comprising the steps 15
- (a) transmitting a third command from a second program to said interface; and
- (b) said interface selectively sending a fourth command representative of said third command to one of said plurality of digital command stations based upon information contained within at least one of said third and fourth commands
- 12. The method of claim 10 wherein said first program and said interface are operating on the same computer.
- 13. The method of claim 11 wherein said first program, said second program, and said interface are all operating on different computers.
- 14. The method of claim 10, further comprising the step of providing an acknowledgment to said first program in 30 response to receiving said first command by said interface prior to sending said second command to one of said plurality of said digital command stations.
- 15. The method of claim 10 wherein said interface communicates in an asynchronous manner with said first pro- 35 gram while communicating in a synchronous manner with said plurality of digital command stations.
- 16. A method of operating a digitally controlled model railroad comprising the steps of:
 - (a) transmitting a first command from a first program to an 40 of: interface:
 - (b) transmitting a second command from a second program to said interface; and
 - (c) said interface sending a third and fourth command representative of said first command and said second command, respectively, to the same digital command station.
- 17. The method of claim 16 wherein said interface communicates in an asynchronous manner with said first and second programs while communicating in a synchronous 50 manner with said digital command station.
- 18. The method of claim 16, further comprising the step of providing an acknowledgment to said first program in response to receiving said first command by said interface prior to sending said third command to said digital command 55 station.
- 19. A method of operating a digitally controlled model railroad comprising the steps of:
 - (a) transmitting a first command from a first program to a first processor; and
 - (b) said first processor providing an acknowledgment to said first program indicating that said first command has properly executed prior to execution of commands related to said first command by said digitally controlled model railroad.
- 20. The method of claim 19, further comprising the step of sending said first command to a second processor which

42 processes said first command into a state suitable for a digital command station.

- 21. The method of claim 19, further comprising the steps of:
 - (a) transmitting a second command from a second program to said first processor; and
 - (b) said first processor selectively providing an acknowledgment to said second program indicating that said second command has properly executed prior to execution of commands related to said second command by said digitally controlled model railroad.
- 22. The method of claim 21, further comprising the steps
- (a) sending a third command representative of said first command to one of a plurality of digital command stations based upon information contained within at least one of said first and third commands; and
- (b) sending a fourth command representative of said second command to one of said plurality of digital command stations based upon information contained within at least one of said second and fourth commands
- 23. A method of operating a digitally controlled model railroad comprising the steps of:
- (a) transmitting a first command from a first program to an asynchronous command processor;
- (b) said asynchronous command processor providing an acknowledgment to said first program indicating that said first command has properly executed prior to execution of said first command by said digitally controlled model railroad;
- (c) sending said first command to a command queue where said asynchronous command processor considers the intended destination device of said first com-
- (d) processing said first command by said synchronous command processor into a suitable format for execution by a digital command station for said digitally controlled model railroad.
- 24. The method of claim 23 further comprising the steps
 - (a) receiving responses from said digital command station; and
 - (b) updating a first database of the state of said digitally controlled model railroad based upon said responses from said digital command station.
- 25. The method of claim 24, further comprising the steps of:
- (a) sending a first response to said command queue from said synchronous command processor where said synchronous command processor considers said command queue the intended destination device of said first response; and
- (b) processing said first response by said asynchronous command processor into a suitable format for said first program.
- 26. The method of claim 25, further comprising the step of updating a second database of the state of said digitally controlled model railroad by said asynchronous command processor based upon said first response from said synchro-60 nous command processor.
 - 27. The method of claim 26, further comprising the step of querying said second database by said asynchronous command processor providing said acknowledgment to said first program providing the information requested and not sending said first command to said command queue.

UNITED STATES PATENT AND TRADEMARK OFFICE **CERTIFICATE OF CORRECTION**

PATENT NO.

: 6,530,329 B2

DATED

: March 11, 2003

INVENTOR(S) : Katzer

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2,

Line 38, change "model railroad In" to -- model railroad. In --

Signed and Sealed this

Fifteenth Day of March, 2005

JON W. DUDAS

Director of the United States Patent and Trademark Office