CAN-open protocol

A Controller Area Network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications withouta host computerIt is a message-based protocol, designed originally for multiplex electrical wiring within automobiles, butis also used in many other contexts.

CANopen is a communication protocol and device profile specification for embedded systemsused in automation.In terms of the OSI model,CANopen implements the layers above and including the network layer. The CANopen standard consists ofan addressing scheme, severalsmallcommunication protocols and an application layerdefined by a device profile.The communication protocols have support for network management, device monitoring and communication between nodes, including a simple transport layer for message segmentation/desegmentation. The lower levelprotocolimplementing the data link and physical layers is usually ControllerArea Network (CAN), although devicesusing some othermeansof communication (such as EthernetPowerlink,EtherCAT) can also implement the CANopen device profile.

Use and benefits of CANopen

CANopen protocol allows multiple controllers to be connected into an extensible unified
network.Its flexible configuration capabilities offer easy access to exposed device parameters and real-time automatic (cyclic or event-driven) data transfer.
The benefits of CANopen include:

•Standardized in EN50325-4
•Widely supported and vendor independent
•Highly extensible
•Offers flexible structure (can be used in a wide variety of application areas)
•Suitable for decentralized architectures
•Wide support of CANopen monitoring tools and solutions

Common Configurations

CAN Mode: Used to select one of the 3 operating modes. Off disables all CAN receive and transmit capabilities.

Node ID: CAN Node ID used for transmission from the controller. Value may be between 1 and 126 included.

Bit Rate: Selectable bit rate. Available speeds are 1000, 800, 500, 250, 125, 50, 25, 10 kbit/s.Default is 125kbit and is the recommended speed for RawCAN and MiniCAN modes.

Heartbeat: Period atwhich a Heartbeat frame is sent by the controller. The

frame is CANopen compatible 0x700 + NodeID,with one data byte of value 0x05 (Status: Operational).The Heartbeat is sent in any of the selected modes. It can be disabled by entering a value of 0.

Supported CAN Modes

Three CAN operating modes are available on the CAN-enabled controllers:

1 - RawCAN

2 - MiniCAN

3 - CANopen

RawCAN is a low-leveloperating mode giving read and write access to CAN frames. It is recommended for use in low data rate systems that do not obey to any specific standard.

CAN framesare typically built and decoded using the MicroBasic scripting language.

MiniCAN is greatly simplified subset of CANopen, allowing, within limits, the integration of the controller into an existing CANopen network. This mode requires MicroBasic scripting to prepare and use the CAN data.

CANopen is the full Standard from CAN in Automation (CIA), based on the DS302 specifcation. It is the mode to use if full compliance with the CANopen standard is a primary requisite.

This section describesthe RawCAN and MiniCAN modes,refer to section “CANopen Interface” for a description of the CANopen mode.

CAN Mode: Used to select one of the 3 operating modes. Off disables all CAN receive and transmit capabilities.

Node ID: CAN Node ID used for transmission from the controller. Value may be between 1 and 126 included.

Bit Rate: Selectable bit rate. Available speeds are 1000, 800, 500, 250, 125, 50,25, 10 kbit/s. Defaultis 125kbitand is the recommended speed for RawCAN and MiniCAN modes.

Heartbeat: Period atwhich a Heartbeat frame is sent by the controller. The frame is CANopen compatible 0x700 + NodeID, with one data byte of value 0x05 (Status: Operational).The Heartbeat is sent in any of the selected modes. It can be disabled by entering a value of 0.

CAN Connection

Connection to a CAN bus is as simple as shown on the diagram above. 120 Ohm Termination Resistors must be inserted at both ends of the bus cable. CAN network can be up to 1000m long. See CAN specifications for maximum length at the various bit rates.

CAN Connection

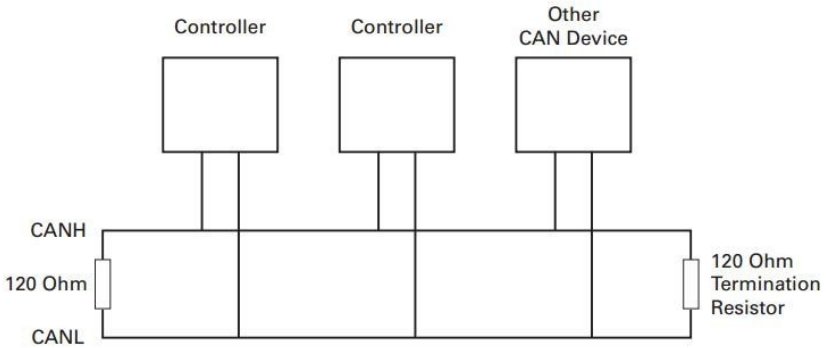

FIGURE 60. CAN connection

CAN Bus Configuration

Though KeyaMotorMonitor software, The controller configures the communication parameters Must be though RS232 to PC.

Use the CAN menu in the Configuration tab in order to enable the CANopen mode. Additionally, the utility can be used to configure the following parameters:

Default configurationL:

•Standard frame
•CAN type: CANopen
•Node ID: 1
•Bit rate:250 kbps
•Heartbeat (ms):1000ms
•Autostart:Enable
•TPDO Enable and Send rate:1500ms

CAN Bus Pinout

Depending on the controller model, the CAN signals are located on the 15-pin female connector or 9-pin male connector. Refer to datasheet for details.

CAN Signals on DB15 connector

Pin Number	Signal	Description
6	CAN_L	CAN bus low
7	CAN_H	CAN bue high

CAN Signals on DB25 connector

Pin Number	Signal	Description
23	CAN_H	CAN bus HIGH
24	CAN_L	CAN bus LOW

CAN and USB Limitations

On most controller models CAN and USB cannot operate at the same time. On

controllers equipped with a USB connector, if simultaneous connection is not allowed, the controller will enable CAN if USB is not connected.

The controller will automatically enable USB and disable CAN as soon as the USB is connected to the PC.The CAN connection will then remain disabled untilthe controller is restarted with the USB unplugged.

See the controller model datasheet to verify whether simultaneous CAN and USB is supported

The controller operating in the CANopen mode can accept the following types of messages:

•Service Data Objects, or SDO messages to read/write parameter values

• Process Data Objects, or PDO mapped messages to automatically transmit parameters and/or accept commands at runtime

•Network Management, or NMT as defined in the CANopen specification

Service Data Object (SDO) Read/Write Messages

Runtime queriesand runtime commands can be sent to the controller in real-time using the expedited SDO messages.

SDO messages provide generic access to Object Dictionary and can be used for obtaining parameter values on an irregular basis due to the excessive network traffic that is generated with each SDO request and response message.

Transmit Process Data Object (TPDO) Messages

Transmit PDO (TPDO) messages are one of the two types of PDO messages that are used during operation.

TPDOsare runtime operating parameters that are sent automatically on a periodic basis from the controller to one or multiple nodes. TPDOs do not alter object data; they only read internal controller values and transmit them to the CAN bus.

Receive Process Data Object (RPDO) Messages

RPDOs are configured to capture runtime data destined to the controller

CANopen implementation supports RPDOs. Data received using RPDOs are

stored in 8 user variables from where they can be processed using MicroBasic scripting.

CANopen data transmitting and receiving

Use CANopen to send and receive data , is not a simple specified ID to send commands and parameters, need to meet certain load specification.

Read node Date

The frame

CANopen compatible 0x700 + NodeID,(0x01-0x7e) Range :0x601-0x67E

Data_0	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6	Data_7
0x40	index	sub-index	00	00	00	00	

The parameters of the target node returns according to the frame format

Data_0	Data_1 2	Data_3	Data_4	Data_5	Data_6	Data_7
0x4F	index	sub-index	XX	00	00	00
0x4B	index	sub-index	XX	XX	00	00
0x47	index	sub-index	XX	XX	XX	00
0x43	index	sub-index	XX	XX	XX	XX

Read Node parameter

Loading ways ofthe ID: 0 x600 + target address ID (0 x01-0 x7e),range of 0x601-0x67E

Loading ways of the data frame is as follows:

Data_0	Data_1 2	Data_3	Data_4	Data_5	Data_6	Data_7
0x2F	index	sub-index	XX	00	00	00
0x2B	index	sub-index	XX	XX	00	00
0x27	index	sub-index	XX	XX	XX	00
0x23	index	sub-index	XX	XX	XX	XX

Note: 0 x2F means message sending a byte of data effectively, and so on, 0 x23 means four bytes of data packets sent.

The target node returned by the CAN data frame format

Data_0	Data_1 2	Data_3	Data_4	Data_5	Data_6	Data_7
0x60	index	sub-index	00	00	00	00

Note: This script does not check for loss of communication on the CAN bus. It is provided for information only

Note:1. The "XX" said one byte of data effectively."00" said invalid bytes, the data is not standard.

2. The CANopen sending and receiving data fields are eight bytes, extra byte is invalid.

3. Multi-byte index, data are low byte in the front.

4. Send invalid command will return an error frame, knowledge is not in the scope of this agreement is introduced.

(3) command example

1. Speed open control
Explain: Instruction index for0x2000 Speed value range(-1000---1000), to 1000 is forward max speed to 0 is output is 0 ,to -1000 is reverse max speed
If you want to control with ID 1 controller, the speed of the open-loop mode, need to send the message as below
ID :0x601
Data:

Speed value of 500 （0x01F4）

0x23	0x00	0x20	0x01	0xf4	0x01	0x00	0x00

Speed value of 1000（0x03E8）

0x23	0x00	0x20	0x01	0Xe8	0x03	0x00	0x00

Speed value of-500 （0x0FE0C，即 0x01F4的补码+1）

0x23	0x00	0x20	0x01	0x0C	0xFE	0x00	0x00

Speed value of -1000 （0x0FC18，即0x03E8的补码+1）

0x23	0x00	0x20	0x01	0x18	0xFC	0x00	0x00

Reply message as below
ID : 0x581
Data :

0x60	0x00	0x20	0x01	0x00	0x00	0x00	0x00

2、Set position mode target value 4000000（0x3D0900）

0x23	0x01	0x20	0x01	0x00	0x09	0x3D	0x00

3、Set position mode speed 1000（0x03E8）

0x23	0x02	0x20	0x01	0xE8	0x03	0x00	0x00

4、Read position counter

0x40	0x04	0x21	0x01	0x00	0x00	0x00	0x00

5、Read run speed

0x40	0x03	0x21	0x01	0x00	0x00	0x00	0x00

6,stop commend

0x2C	0x0E	0x20	0x01	0x01	0x00	0x00	0x00

When Test, it is recommended to use the CAN card to send packets and monitor the bus status.

Object Dictionary

index	sub	Entry Name	Data Type & Access	Command Reference
0x2000	01	Set motor commend,ch 1	S16 WO	“G”
	02	Set motor commend,ch 2		
0x200C	00	Emergency Shutdown	U8 WO	“ EX”
0x200D	00	Release Shutdown	U8 WO	“MG”
0x2C0E	01	Set motor commend,ch 1	U8 WO	“MS”
	02	Set motor commend,ch 2		
0x2017	00	Save Config to Flash	U8 WO	“EES”

Instruction index form

index	sub	Entry Name	Data Type & Access	Command Reference
0x2001	01	Set Position commend, ch.1	S32 WO	“P”
	02	Set Position commend, ch.2		
0x2002	01	Set Position speed, ch.1	S16 WO	“S”
	02	Set Position speed, ch.2		
0x2003	01	Set Encoder Counter, ch.1	S32 WO	“C”

	02	Set Encoder Counter, ch.2		
0x2006	01	Set Acceleration , ch.1	S32 WO	“MAC”
	02	Set Acceleration , ch.2		
0x2007	01	Set Deceleration , ch.1	S32 WO	“ MDEC”
	02	Set Deceleration , ch.2		

Instruction index form

index	sub	Entry Name	Data Type & Access	Command Reference
0x2100	01	Read Motor Amps, ch.1	S16 RO	“A”
	02	Read Motor Amps, ch.2	S16 RO	
0x2101	01	Read Actual Motor Command, ch.1	S16 RO	“ M”
	02	Read Actual Motor Command, ch.2		
0x2103	01	Read Encoder Motor Speed, ch.1	S16 RO	“S”
	02	Read Encoder Motor Speed, ch.2	S16 RO	
0x2107	01	Read Encoder Motor Speed as 1/1000 of Max, ch.1	S16 RO	“SR”
	02	Read Encoder Motor Speed as 1/1000 of Max, ch.2		
0x2108	01	Read Encoder Count Relative, ch.1	S32 RO	“ CR”
	02	Read Encoder Count Relative, ch.2		
0x2109	01	Read Brushless Count Relative, ch.1	S32 RO	“CBR”
	02	Read Brushless Count Relative, ch.2		
0x210A	01	Read BL Motor Speed in RPM, ch.1	S16 RO	“BS”
	02	Read BL Motor Speed in RPM, ch.2		
0x210C	01	Read Battery Amps, ch.1	S16 RO	“ BA”
	02	Read Battery Amps, ch.2		
0x210D	01	Read Internal Voltages (V Int)	U16 RO	“V”
	02	Read battery Voltages (V)	U16 RO	
	03	Read Internal Voltages (mV)	U16 RO	
0x210E	00	Read All Digital Inputs	U32 RO	“D”
0x210F	01	Read Case & Internal Temperatures (MCU Temperature)	S8 RO	“T”
	02	Read Case & Internal Temperatures (ch.1)		
	03	Read Case & Internal Temperatures (ch.2)		
0x2111	00	Read Status Flags	U8 RO	“FS”
0x2112	00	Read Fault Flags	U8 RO	“FF”

0x2113	00	Read Current Digital Outputs	U8 RO	“DO”
0x2119	00	Read time	U32 RO	“TM”
0x2121	00	Read motor Status	U8 RO	“FM”

General agreement details:

EX - Emergency Stop

CANOpen id: 0x200C

Description:

The EX command will cause the controller to enter an emergency stop in the same way as if hardware emergency stop was detected on an input pin. The emergency stop condition will remain until controller is reset or until the MG release command is received.

Syntax Serial: !EX

MG - Emergency Stop Release

CANOpen id: 0x200D

Description:

The MG command willrelease the emergency stop condition and allow the controller toreturn to normal operationAlways make sure that the fault condition has been cleared before sending this command

Syntax Serial: !MG

MS - Stop in all modes

CANOpen id: 0x200E

Description:

The MS command is similar to the EX emergency stop command except that it is applied to the specifed motor channel

Syntax Serial: !MS [cc] Motor Channel

EES - Save Confguration in EEPROM

CANOpen id: 0x2017

Description:

This command causes any changes to the controller’s confguration to be saved to Flash.Saved confgurations are then loaded again next time the controller is powered on. This command isa duplication of the EESAV maintenance command. It is provided as a Real-Time command as well in order to make it possible to save confguration changes from within MicroBasic scripts.

Syntax Serial: !EES

P - Go to Absolute Desired Position

CANOpen id: 0x2001

Description:

This command is used in the Position Count mode to make the motor move to a specifed encoder or hall count value.Runtime Commands

Syntax Serial: !P [cc] nn

cc = Motor channel

nn = Absolute count destination

Example:

!P 1 10000 : make motor go to absolute count value 10000.

S - Set Motor Speed

CANOpen id: 0x2002

Description:

In the Closed-Loop Speed mode, this command will cause the motor to spin at the desired RPM speed. In Closed-Loop Position modes, this commands determines the speed at which the motor will move from one position to the next. It will not actually start the motion.

Syntax Serial: !S [cc] nn

Argument 1: Channel

Min: 1 Max: Total Number of Motors

Argument 2: Value Type: Signed 32-bit

Min: -500000 Max: 500000

C - Set Encoder Counters

CANOpen id: 0x2003

Description:

This command loads the encoder counter for the selected motor channel with the value contained in the command argument. Beware that changing the controller value while operating in closed-loop mode can have adverse effects.

Syntax Serial: !C [cc] nn

cc = Motor channel

nn = Counter value

Example:

!C 2 -1000 : Loads -1000 in encoder counter 2

!C 1 0 : Clears encoder counter 1

AC - Set Acceleration

Alias: ACCEL HexCode: 07 CANOpen id: 0x2006

Description:

Set the rate of speed change during acceleration for a motor channel.This command is identical to the MACC confguration command but is provided so that it can be changed rapidly during motor operation. Acceleration value is in 0.1 * RPM per second. When using controllers ftted with encoder, the speed and acceleration value are actual RPMs. Brushless motor controllers use the hall sensor for measuring actual speed and acceleration will also be in actual RPM/s. When using the controller without speed sensor, the acceleration value is relative to the Max RPM confguration parameter, which itself is a user-provided number for the speed normally expected at full power. Assuming that the Max RPM parameter is set to 1000, and acceleration value of 10000 means that the motor will go from

0 to full speed in exactly 1 second, regardless of the actual motor speed.

Syntax Serial: !AC cc nn

cc = Motor channel

nn = Acceleration value in 0.1 * RPM/s

Example:

!AC 1 2000 : Increase Motor 1 speed by 200 RPM every second if speed is measured by encoder

!AC 2 20000 : Time from 0 to full power is 0.5s if no speed sensors are present and Max RPM is set to 1000

DC - Set Deceleration

CANOpen id: 0x2007

Description:

Set the rate of speed change during decceleration for a motor channel. This command is identical to the MDEC confguration command but is provided so that it can be changed rapidly during motor operation. Decceleration value is in 0.1 * RPM per second. When using controllers ftted with encoder, the speed and decceleration value are actual RPMs.

Brushless motor controllers use the hall sensor for measuring actual speed and decceleration will also be in actual RPM/s. When using the controller without speed sensor, the decceleration value is relative to the Max RPM confguration parameter, which itself is user-provided number for the speed normally expected at full power. Assuming that the Max RPM parameter is set to 1000, and decceleration value of 10000 means thatthe motorwill go from full speed to 0 in exactly 1 second, regardless of the actual motor speed.

Syntax Serial: !DC cc nn

cc = Motor channel

nn = Deceleration value in 0.1 * RPM/s

Example:

!DC 1 2000 : Reduce Motor1 speed by 200 RPM every second ifspeed is measured by encoder

!DC 2 20000 : Time from fullpower to stop is 0.5s ifno speed sensors are present and Max RPM is set to 1000

A - Read Motor Amps

Alias: MOTAMPS HexCode: 00 CANOpen id: 0x2100

Description:

Measures and reports the motor Amps for all operating channels. Note that the current flowing through the motors is often higher than this flowing through the battery.

Syntax Serial: ?A [cc]

cc = Motor channel

aa = Amps *10 for each channel

Example:

Q: ?A

R: A=100:200

Q: ?A 2
R: A=200

V - Read Volts

CANOpen id: 0x210D

Description:

Reports the voltages measured inside the controller at three locations: the main battery voltage, the internal voltage at the motor driver stage, and the voltage that is available on the 5V output on the DSUB 15 or 25 front connector. For safe operation, the driver stage voltage must be above 12V. The 5V output willtypically show the controller's internal regulated 5V minus the drop of a diode that is used for protection and will be in the 4.7V range. The battery voltage is monitored for detecting the undervoltage or overvoltage conditions.

Syntax Serial: ?V [ee]

ee =

1 : Internal volts

2 : Battery volts

3 : 5V output

nn = Volts * 10 for internal and battery volts. Milivolts for 5V output

Example:

Q: ?V

R:V=135:246:4730

Q: ?V 3

R:V=4730

C - Read Encoder Counter Absolute

CANOpen id: 0x2104

Description:

Returns the encoder value as an absolute number. The counter is 32-bit with a range of +/- 2147483648 counts.

Reply:

C=nn Type: Signed 32-bit Min: -2147M Max: 2147M

Where:

cc = Encoder channel number

nn = Absolute counter value

T - Read Temperature

CANOpen id: 0x6403

Description:

Reports the temperature at each of the Heatsink sides and on the internal MCU silicon chip. The reported value is in degrees C with a one degree resolution.

Syntax Serial: ?T [cc]

Reply:

T= cc Type: Signed 8-bitMin: -40 Max: 125

Where:

cc =
1 : MCU temperature
2 : Channel 1 side
3 : Channel 2 side
tt = temperature in degrees
Note:
On some controller models, additional temperature values may reported. These are measured at different points and not documented.You may safely ignore this extra data. Other controller models only have one heatsink temperature sensor and therefore only report one value in addition to the Internal IC temperature.

FS - Read Status Flags

CANOpen id: 0x6405
Description:
Report the state of status flags used by the controller to indicate a number of internal conditions during normal operation. The response to this query is the single number for all status flags.The status of individual flags is read by converting this number to binary and look at various bits of that number.

Syntax Serial: ?FS
FS = f1 + f2*2 + f3*4 + ... + fn*2^n-1 Type: Unsigned 8-bit Min: 0 Max: 255
Where:
f1 = Serial mode
f2 = Pulse mode
f3 = Analog mode
f4 = Power stage off
f5 = Stall detected
f6 = At limit
f7 = Unused
f8 = MicroBasic script running
FF - Read Fault Flags

CANOpen id: 0x6406

Description:
Reportsthe status of the controllerfault conditionsthat can occur during operation. The response to that query is a single number which must be converted into binary in order to evaluate each of the individual status bits that compose it.

Syntax Serial: ?FF
FF = f1 + f2*2 + f3*4 + ... + fn*2n-1
Type: Unsigned 8-bit Min: 0 Max: 255
Where:
f1 = Overheat
f2 = Overvoltage
f3 = Undervoltage

f4 = Short circuit
f5 = Emergency stop
f6 = Brushless sensor fault
f7 = MOSFET failure
f8 = Default confguration loaded at startup
Example:
Q: ?FF
R: FF=2 : Overvoltage fault

FM - Read Runtime Status Flag

Alias: MOTFLAG HexCode: 30 CANOpen id: 0x2122
Description:
Report the runtime status of each motor. The response to that query is a single numberwhich mustbe converted into binary in orderto evaluate each ofthe individual status bits that compose it.
Syntax Serial: ?FM [cc]
cc = Motor channel
f1 = Amps Limit currently active
f2 = Motor stalled
f3 = Loop Error detected
f4 = Safety Stop active
f5 = Forward Limit triggered
f6 = Reverse Limit triggered
f7 = Amps Trigger activated
Example:
Q: ?FM 1
R: FM=6 : Motor 1 is stalled and loop error detected
Note:
On controller models supporting Spektrum radio mode f4 is used to indicate Spektrum. f4 to f6 are shifted to f5 to f7