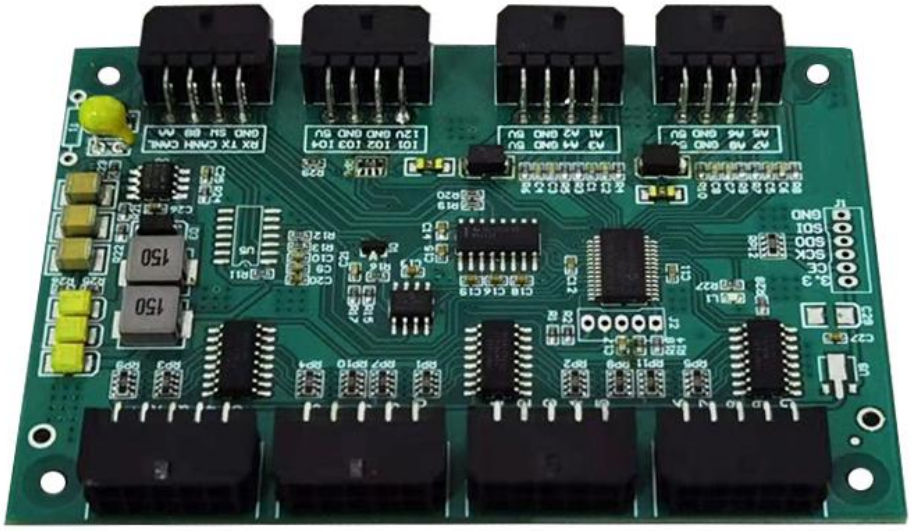
MKF-C100-EB

Mate Emulator Board, CAN communication



DESCRIPTION

CAN communication acquisition board, CAN interface, CAN2.0B protocol standard, supports standard frame ID and extended frame ID, user repairable ID, ID standard of CIA J1939 and CAN OPEN protocol. Support 8-axis analog input, can connect 20 buttons, 3 indicator lights interface, 1 CAN interface, 1 RS232 interface. It is suitable for the control panel of CNA communication to quickly customize the control panel of CAN communication for customers.

FEATURES

- Analog input: 8-channel 0-5V signal input, 12-bit precision
- IO input: 20 (external button or switch)
- Indicator light output: 1 power light, 1 function indicator, 2 programmable LED indicators
- Communication interface: CAN2.0 or RS232
- Dimensions: (W)100X(L)70X(H)12
- Positioning hole size: (W)90X(L)60, through hole Ø3.0mmX4
- CAN parameters can be set (CAN communication parameters can be set through the RS232 interface)
- Support standard frame ID, extended frame ID and remote frame
- Button interface: 1-20, connected to the button, the upper and lower wires of each port are respectively connected to the 2 terminals of the button
- RS232 interface: TX data transmission, RX data reception, GND ground
- CAN interface: CAN-L white, CAN-H blue

ACCESSORIES

- 3.0-pitch 8P wire, 30CM length, 4 pieces
- 3.0-pitch 10P wire, 4 pieces

MATE TECHNOLOGY SHENZHEN LIMITED
info@matend.com | www.matend.com | www.matend.cn

BASIC PARAMETERS

Hardware configuration, 8 channel analog quantity, 20 switch quantity
CAN ID can be set arbitrarily
CAN baud rate can be set arbitrarily
Communication method: regular sending, sending interval can be set arbitrarily

INTERPRETATION OF ANALOG DATA

XX (Hexadecimal): Analog data (0x00-0xFF)
0X00-0X7F Reverse
0X80 Stop
0X81-0XFF Forward

SWITCH DATA INTERPRETATION

Button 1: button group 1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 8	Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1

Button 2: button group 2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 16	Button 15	Button 14	Button 13	Button 12	Button 11	Button 10	Button 9

Button 3: button group 2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				Button 20	Button 19	Button 18	Button 17

Button press=1, release=0
Short circuit between button and GND = ON
Open circuit OFF

CAN COMMUNICATION DATA FORMAT

1. Protocol=00: 8-channel analog quantity, no switching quantity

Data frame (8 bytes HEX):

BYTE0	A1	Analog value 1	(0x00-0x80-0xff)
BYTE1	A2	Analog value 2	(0x00-0x80-0xff)
BYTE2	A3	Analog value 3	(0x00-0x80-0xff)
BYTE3	A4	Analog value 4	(0x00-0x80-0xff)
BYTE4	A5	Analog value 5	(0x00-0x80-0xff)
BYTE5	A6	Analog value 6	(0x00-0x80-0xff)
BYTE6	A7	Analog value 7	(0x00-0x80-0xff)
BYTE7	A8	Analog value 8	(0x00-0x80-0xff)

2. Protocol=01: 7-channel analog quantity, 8 switch quantity

Data frame (8 bytes HEX):

BYTE0	A1	Analog value 1	(0x00-0x80-0xff)
BYTE1	A2	Analog value 2	(0x00-0x80-0xff)
BYTE2	A3	Analog value 3	(0x00-0x80-0xff)
BYTE3	A4	Analog value 4	(0x00-0x80-0xff)
BYTE4	A5	Analog value 5	(0x00-0x80-0xff)
BYTE5	A6	Analog value 6	(0x00-0x80-0xff)
BYTE6	A7	Analog value 7	(0x00-0x80-0xff)
BYTE7	Button 1	Button 1-8	1 = ON, 0 = OFF

3. Protocol=02: 6-channel analog quantity, 16 switch quantity

Data frame (8 bytes HEX):

BYTE0	A1	Analog value 1	(0x00-0x80-0xff)
BYTE1	A2	Analog value 2	(0x00-0x80-0xff)
BYTE2	A3	Analog value 3	(0x00-0x80-0xff)
BYTE3	A4	Analog value 4	(0x00-0x80-0xff)
BYTE4	A5	Analog value 5	(0x00-0x80-0xff)
BYTE5	A6	Analog value 6	(0x00-0x80-0xff)
BYTE6	Button 1	Button 1-8	1 = ON, 0 = OFF
BYTE7	Button 2	Button 9-16	1 = ON, 0 = OFF

4. Protocol=03: 5-channel analog quantity, 20 switch quantity

Data frame (8 bytes HEX):

BYTE0	A1	Analog value 1	(0x00-0x80-0xff)
BYTE1	A2	Analog value 2	(0x00-0x80-0xff)
BYTE2	A3	Analog value 3	(0x00-0x80-0xff)
BYTE3	A4	Analog value 4	(0x00-0x80-0xff)
BYTE4	A5	Analog value 5	(0x00-0x80-0xff)
BYTE5	Button 1	Button 1-8	1 = ON, 0 = OFF
BYTE6	Button 2	Button 9-16	1 = ON, 0 = OFF
BYTE7	Button 3	Button 17-24	1 = ON, 0 = OFF

RS232 COMMNICATION DATA FORMAT

Baud rate: 115200, 1 start bit, 8 data bits, 1 stop bit, no parity bit
Data format sent by joystick: (control panel-> PC) (13 bytes HEX):

BYTE0	0xFF	Header	
BYTE1	X	Joystick X axis	X axis(00-FF) A1
BYTE2	Y	Joystick Y axis	Y axis(00-FF) A2
BYTE3	Z	Joystick Z axis	Z axis(00-FF) A3
BYTE4	T	RX axis	T axis (00-FF) A4
BYTE5	A	RY axis	A axis (00-FF) A5
BYTE6	B	RZ axis	B axis (00-FF) A6
BYTE7	C	The 7 axis	C axis (00-FF) A7
BYTE8	D	The 8 axis	D axis (00-FF) A8
BYTE9	Button 1	Button 1-8	(00-FF) HEX
BYTE10	Button 2	Button 9-16	(00-FF) HEX
BYTE11	Button 3	Button 17-20	(00-FF) HEX
BYTE12	CH	Checksum	(00-FF) HEX

Checksum BYTE12(CH)= BYTE1+ BYTE2+ BYTE3+ BYTE4+ BYTE5+…… BYTE11 The low byte of the sum of all bytes except the header

Eg. FF 80 80 80 80 80 80 80 82 00 00 00 02

Button1: Button group 1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 8	Button 7	Button 6	Button 5	Button 4	Button 3	Button 2	Button 1

Button 2: button group 2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Button 16	Button 15	Button 14	Button 13	Button 12	Button 11	Button 10	Button 9

Button 3: button group 2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SW	Button 20	Button 19	Button 18	Button 17

Button press=1, release=0

1. Set the axis parameters:

Format: af 0C aa bb 00 00 f5

A. aa is the axis parameter, mainly to set automatic return to center or not to return to center (whether center calibration)

Bit0-Bit7 correspond to A1-A8 (8 axes), which are binary parameters: 0000 0111 (07)

1=Spring automatic return (rocker), 0=Friction resistance positioning (such as knobs, push rods, etc.)

Default: all 8 axes are automatically centered

For example: A1, A2, A3 correspond to a 3-axis joystick XYZ, and A4~A8 are connected to potentiometers. aa=0x07

All 8 axes are automatically centered: aa=0xff (default)

B. bb function: 4 CAN data commands, Bit3,2,1,0 correspond to CAN data commands 4,3,2,1

1=Send this command

0=Do not send this command

For example:

bb=00001111 Send data 1, data 2, data 3, data 4 in sequence during CAN transmission

bb=00000001 Send data 1 in sequence during CAN transmission (others are not sent)

eg: af 0C 07 01 00 00 f5

2. Communication refresh rate (10-100MS):

Format: af 11 XX 00 00 00 f5

X X is the refresh rate, the unit is MS, which refers to how long it takes to send a frame of data;

Data valid interval 10~100ms such as setting 25ms (0x19)

25ms: af 11 19 00 00 00 f5

33ms: af 11 21 00 00 00 f5

45ms: af 11 2d 00 00 00 f5

3. Communication port:

Format: af 05 XX 00 00 00 f5 XX is the setting parameter, the details are as follows;

XX: 00=CAN, 01=RS232

CAN: af 05 00 00 00 00 f5 (default)

RS232: af 05 01 00 00 00 f5

4. RS232 baud rate:

Format: af 0B XX 00 00 00 f5 XX is the setting parameter, as follows;

9600: af 0B 00 00 00 00 f5 (default)

19200: af 0B 01 00 00 00 f5

57600: af 0B 02 00 00 00 f5

115200: af 0B 03 00 00 00 f5

5. Communication mode (main mode automatically sends at regular time, slave mode query) (RS232 setting PC->Joystick)

Slave mode: The joystick is a slave device. Only after receiving a query command from the host, will it send data back to the host.

Main mode: Timing and automatic sending: The joystick will automatically send data to the host at a fixed time when the joystick is powered on. For the sending rate, refer to "Refresh Rate Setting"

This parameter joystick is always stored (MATE has been set for the customer)

Format:

0xaf	0x08	00	00	00	Mode	0xf5
Header	Command	Data 1	Data 2	Data 3	Data 4	Tail

Mode=00 master mode, Mode=01 slave mode

For example: (Set via RS232 port, PC->Joystick)

Master mode (timing transmission) af 08 00 00 00 00 f5 (HEX)

Slave mode (query method) af 08 00 00 00 01 f5 (HEX)

After the setting is successful, the joystick returns to ACK (AA 55 AF) (Joystick->PC)

6. Set the center point: Format: af 09 00 00 00 00 f5

7. Query status:

Query format: af 07 00 00 00 00 f5

After sending this command once, a 21-byte status message will be returned. If the control board is not sent, there will be no return sign. The automatic sending mode can be restored after power off and restart.

8. Reset joystick:

Format: af 15 00 00 00 00 f5

9. Device information query:

Query format: af 20 00 00 00 00 f5

Return format: AA name,year,month,day, ComPort, Baud, Refresh, AxisOperation, 55

For example, return: AA A8 18 06 05 00 03 19 FF 55

10. CAN port baud rate: (PC->Joystick)

0xaf 0x06 XX 00 00 00 0xf5

Head command data 1 Data 2 Data 3 Data 4 Tail

XX=00 125K

XX=01 250K (default)

XX=02 500K

XX=03 1000K

For example:
af 06 00 00 00 00 f5 (HEX) CAN baud rate=125K
af 06 01 00 00 00 f5 (HEX) CAN baud rate=250K (default)
af 06 02 00 00 00 f5 (HEX) CAN baud Rate=500K
af 06 03 00 00 00 f5 (HEX) CAN baud rate=1000K

11. Joystick "CAN ID" setting: the ID of the data sent by the joystick (set via RS232, PC->Joystick) Only applicable to "common protocol", CANopen protocol does not use this command

0xaf	0x01	D1	D2	D3	D4	0xf5
Head	command	Data 1	Data 2	Data 3	Data 4	Tail

D1.7=0 Extended frame 29 bits
D1.7=1 Standard frame 11 bits

- 29-bit extended frame: data range 0X0-0X0FFFFFFF, data D1-D4 correspond to "node identification code"

  For example: Set the sending node identification code-extended frame "0X00F0F101"
  af 01 00 f0 f1 01 f5 (HEX)
- 11-bit standard frame: data range 0X000-0X3FF, data D3-D4 corresponds to "node identification code"
  For example: Set the sending node identification code-standard frame "0X181"
  af 01 80 00 01 81 f5 (HEX)

12. CAN protocol setting:: (RS232 setting PC->Joystick) Mate has helped the customer to set it up

0xaf	0x0A	00	00	00	SS	0xf5
Header	Command	Data 1	Data 2	Data 3	Data 4	Tail

SS refers to the protocol format
SS=00 8 analog quantity Input without switching value: af 0a 00 00 00 00 f5 (HEX)
SS=01 7 analog input, 8 switching values: af 0a 00 00 00 01 f5 (HEX)
SS=02 6 analog input, 16 switches Quantity: af 0a 00 00 00 02 f5 (HEX)
SS=03 5 analog input, 20 switching values: af 0a 00 00 00 03 f5 (HEX)

13. Query (remote frame) "CAN ID" setting: refers to the frame ID that the joystick can receive.
If the received ID is the same as this ID, then this command will be received and the corresponding processing will be performed.
Suitable for query mode, host CAN ID (remote frame) is set via RS232, (PC->Joystick)

0xaf	0x02	D1	D2	D3	D4	0xf5
Head	command	Data 1	Data 2	Data 3	Data 4	Tail

D1.7=0 Extended frame 29 bits, D1.7=1 standard frame 11 bits

29-bit extended frame: data range 0X0-0X0FFFFFFF, data D1-D4 correspond to "node identification code"
For example: Set the receiving node identification code-extended frame "0X00F0F101"
af 02 00 f0 f1 01 f5 (HEX)

11-bit standard frame: data range 0X000-0X3FF, data D3-D4 corresponds to "node identification code"
For example: Set the identification code of the receiving node-standard frame "0X1E1"
af 02 80 00 01 E1 f5 (HEX)