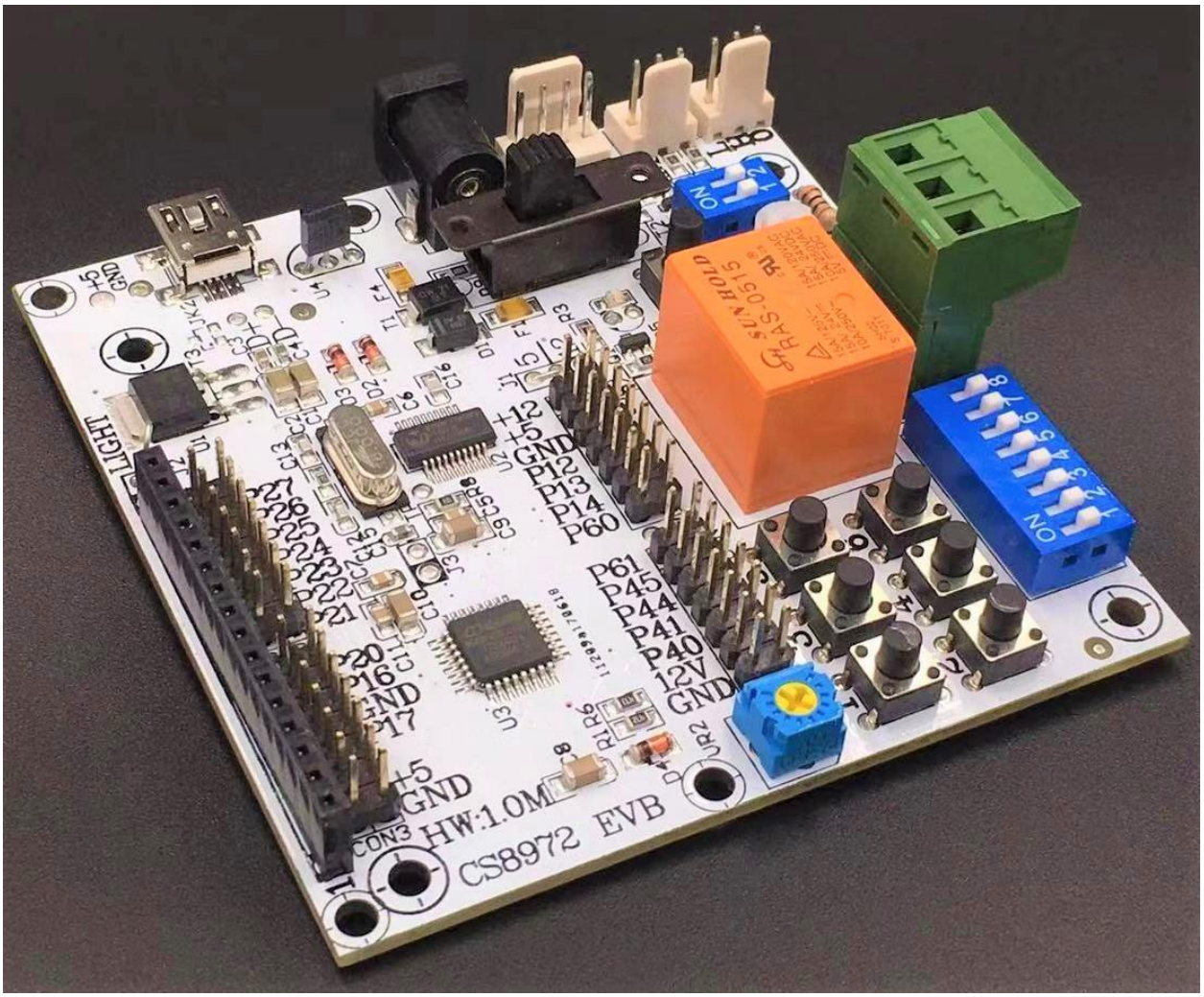
CS8972 EVB

(SWI-CAN-CS-8972-202006)

User Manual v1.0



2020/06

SwiSys Co., Ltd.
Tainan, Taiwan.

Noted

- **This document is intended for the 「CS8972 EVB User Manual」. Some of the information, such as the engineering specification or BOM or firmware source codes, described in this document should not appear in any of the 「EVB User Manual」.**
-

Table of Contents

List of Figures

Address: No.2, Aly. 26, Ln. 442, Kaiyuan Rd., North Dist., Tainan City 704, Taiwan (R.O.C.)
Tel: +886-6-200-6070 Fax: +886-6-275-9435 Website: http://swisystem.com/

1. Introduction

The CS8972 EVB is a stand-alone Controller Area Network (CAN) protocol controller with the embedded CAN transceiver. The CAN transceiver meets or exceeds ISO 11898 standards. As CAN transceivers, these devices provide differential transmit and receive capability as signaling rates up to 1 Mb/s for a CAN controller. It is capable of transmitting and receiving standard and extended message frames. It includes eight independent auto-dispatch and 1024-byte transmit buffers, FIFO receives filtering with 12 ID acceptance and message management. The MCU communication is implemented via an industry standard Serial Peripheral Interface (SPI) and an I^2C bus.

2. Product Description

- The CS8972 is a stand-alone CAN protocol controller with the embedded CAN transceiver.
- It supports standard CAN 2.0B, and the maximal bit rate is 1 Mb/s.
- CS8972 provides two popular serial interfaces for external MCU communication, Serial Peripheral Interface (SPI) and an I^2C bus.
- SPI and I^2C are both slave controllers in CS8972. They receive and respond through the selected serial interface to MCU commands.
- In CS8972, MCU only implements a slave SPI controller to receive and respond to the commands.
- CS8972 has a power saving mode when the external oscillator (XOSC) and internal oscillator (IOSC) are switched OFF. Built-in IOSC up to 16MHz.

3. Main Functions

The CAN controller is compatible with CAN 2.0A and 2.0B standards. The controller can be configured as a normal operating mode or listen-only mode. A self-test loop-back mode can also be enabled to perform internal loop-back test of the CAN controller. The receiver includes four acceptance filters that are used for ID filtering. The matched messages are stored in receive FIFO shared with CPU XRAM. The transmitter includes a 13-byte transmit buffer in XFR consisting of frame information, 4 bytes of message ID, and 8 bytes of message data. The transmitter also includes three dispatchers. The dispatcher is used to automatically transmit a pre-determined message at a fixed programmable time interval without CPU intervention. Each dispatcher also includes a 13-byte transmit buffer in XFR similar to the main transmit path.

After reset, the CAN controller is put in reset mode with all state-machines forced in the initialization states. The user must exit this reset mode by setting CANRST=0 to enter into operating modes.

In CS8972, the CAN controller has the main function as follows:

- ❖ Operating mode
- ❖ Reset mode
- ❖ Listen-Only mode

- Bus Auto-Recovery
- Self-test mode
- Self-reception
- Programmable Baud-Rate
- 8 transmit buffers for standard and extended message frames
- Transmit dispatch
- 8 acceptance filter and mask for receiving frame
- Error detection
- Loop test

4. Specifications

- Single + 5V power supply
- Maximal operating clock up to 24MHz
- Built-in one CAN controller according to CAN protocol version 2.0B.
  - 0 to 8-byte message length
  - Standard and extended data frames
  - Programmable bit rate up to 1Mb/s
  - Support for remote frames
  - 1024 Bytes receive FIFO
  - 12 full acceptance filters
  - 8 full filter masks
  - 8 transmit buffers with abort feature
  - Auto-dispatch function for each transmit buffer
  - Listen mode
  - Loop-back mode for self-test operation
- Built-in CAN transceiver full support CAN v2.0B specification
  - Built-in IOSC up to 16MHz
  - Hardware interface
  - High speed SPI interface up to 3Mb/s bit rate
  - Supports SPI mode 0,0 and 1,1
  - High speed Slave I2C interface up to 2Mb/s bit rate
  - Interrupt output pin with selectable enables
- Low power CMOS technology
  - 50mA active current typical
  - 1mA standby current
- External wake-up by SPI/ I2C and CAN receiver
- Industrial operating temperature range (-40°C to +125°C)
- 20-pin SSOP packages
- RoHS compliance

5. CS8972 Evaluation Board

Slave Board (Top View)

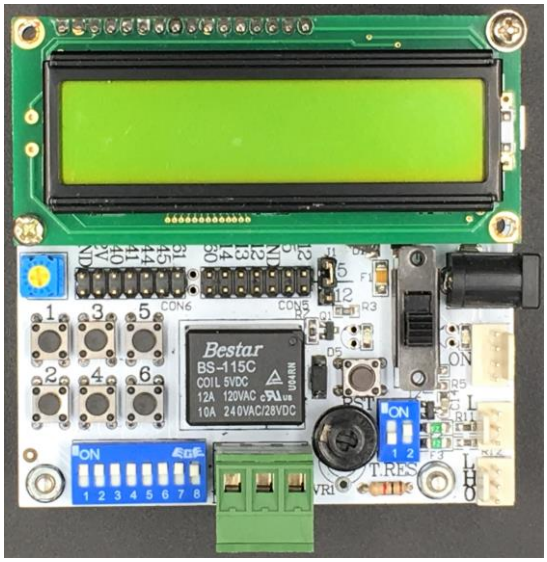


Figure 1: CS8972 Slave Board

Slave Board (Layout)

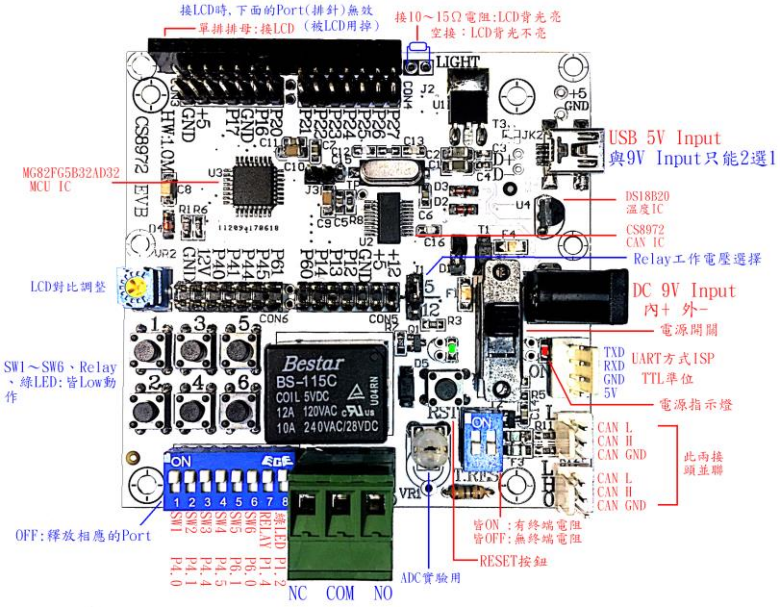


Figure 2: CS8972 Slave Board Layout

6. CS8972 PCB Layout

The PCB layout for CS8972 EVB is shown in the Figure 3.

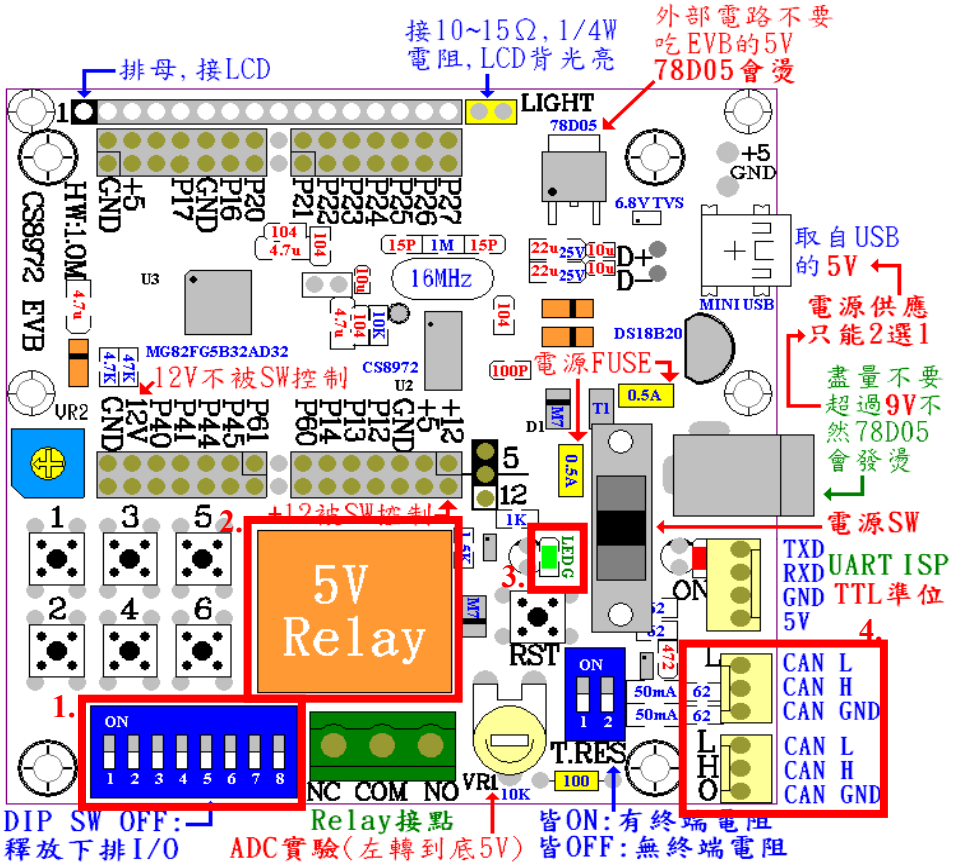


Figure 3: CS8972 PCB Layout

- Description (Numbers shown as a red box)
  1. The 8 DIP SW I/O Pins details:
     - ❖ GND ⟷ SW1 ⟷ DIP SW1 → P4.0 (Low action)
     - ❖ GND ⟷ SW2 ⟷ DIP SW2 → P4.1 (Low action)
     - ❖ GND ⟷ SW3 ⟷ DIP SW3 → P4.4 (Low action)
     - ❖ GND ⟷ SW4 ⟷ DIP SW4 → P4.5 (Low action)
     - ❖ GND ⟷ SW5 ⟷ DIP SW5 → P6.1 (Low action)
     - ❖ GND ⟷ SW6 ⟷ DIP SW6 → P6.0 (Low action)
  2. Relay ⟷ DIP SW6 → P1.4 (Low action)
  3. Green LED ⟷ DIP SW8 → P1.2 (Low action)
  4. 2 x 3 CAN Bus pin connectors in parallel to access various groups

7. LCD and MCU PIN Definition

- LCD Pin such as Receive Status, Enable and Data registers (D0-D7). The MCU Pin such as P1.7.P1.6,P2.0,P2.1,P2.2,P2.3,P2.4,P2.5,P2.6 and P2.7. Table 1 displays the DIP SW pin, SW position and Pin description in CS8972.

Table 1: LCD Pin, MCU Pin, DIP SW Pin SW position and Pin description in CS8972.

LCD Pin	RS	EN	D0	D1	D2	D3	D4	D5	D6	D7
MCU Pin	P1.7	P1.6	P2.0	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6	P2.7

DIP SW Pin	1	2	3	4	5	6	7	8
SW Position	SW1	SW2	SW3	SW4	SW5	SW6	Relay	Green LED
MCU Pin	P4.0	P4.1	P4.4	P4.5	P6.1	P6.0	P1.4	P1.2

- DIP SW in OFF mode:

  If DIP SW OFF, the corresponding MCU Pin component will be released.

Pin Description	VR1	UNUSED	U4 TEMPERATURE IC	U2 MOSI	U2 SCS\	U2 MISO	U2 SCLK
MCU Pin	P1.1	P1.3	P1.0	P3.3	P3.4	P3.5	P1.5

- ADC 1 is connected in P1.1 in MCU.

8. ISP Tool Settings via COM port

- First, confirm all power shut down in『CS8972』EVB Board and connect GND, TXD, RXD (all TTL levels).
- A user open Megawin COM Port ISP version 2.90 is shown in Figure 4.
- Select MCU Type as 82 series : MG82FG5B32.
- Load the program in the format of AP (code).

Address: No.2, Aly. 26, Ln. 442, Kaiyuan Rd., North Dist., Tainan City 704, Taiwan (R.O.C.)
Tel: +886-6-200-6070 Fax: +886-6-275-9435 Website: http://swisystem.com/

- Press【Update Target】and switch on the power supply from 『CS8972』 EVB.
- If not, verify 【Update Target】confirm that the COM Port is correct. Repeat the above actions.

Note：If the UART ISP uses USB to UART cable and is powered by the cable, it is necessary to connect the Mini USB and DC power supply JACK (only one power source) (5V of Mini USB and 5V of ISP are connected).

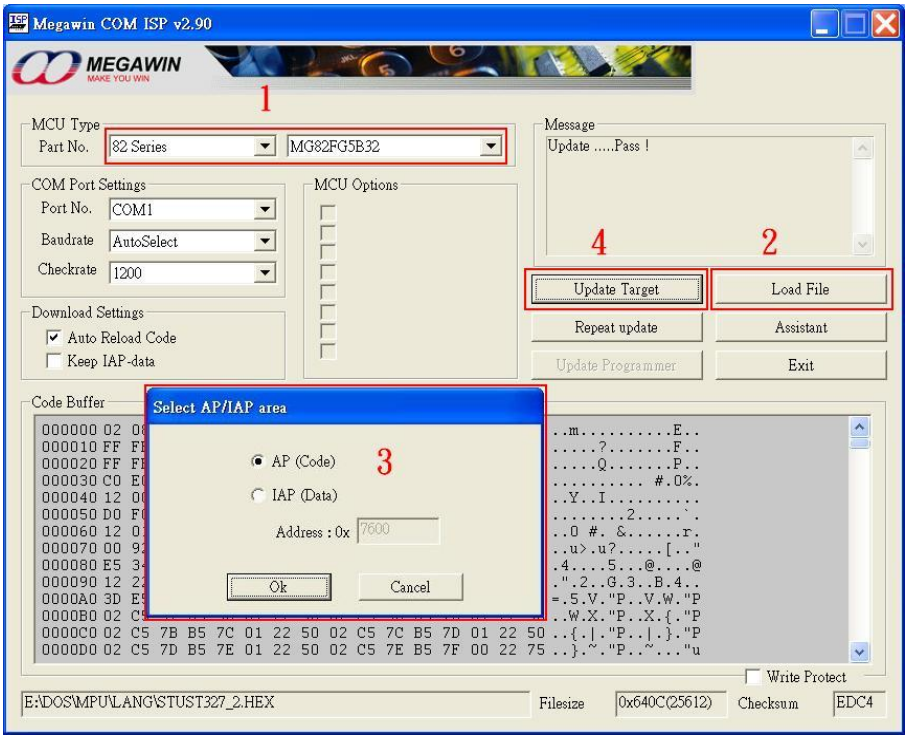


Figure 4: Megawin COM ISP v2.90

9. Instruction Manual

9.1. Initialize CS8972

- Make sure the CS8972 register is in RESET mode and that CANMODE(00h) bit_0 (RSTM) is set to 1. Some registers in RESET mode.
- Set and enabled system clock source (recommended to choose an external crystal oscillator) (IOCTL (78h) is written 0Ch), Need to unlock before doing this action unlock (Write A6h to CAN protection lock register (Protection Locker Register: 7Fh)).
- Configure CAN BUS Bit Rate (09h and 0Ah)(Example 500K: 09h write 40h, 0Ah write 3Ah).
- Set the mask bits of the acceptance filter mask register 0 (50h ~ 53h) (with 8 groups) are logic 1. (This example is received regardless of the ID).
- If you want to receive CAN packets with a specific ID, you need to set a fully matched acceptance filter (12h from 4h to 4Fh).
- Make it possible to fully match device recognition and mask filter 0: 70h write 01h, 71h write 00h, and enable its interrupt: 72h write 01h, 73h write 00h (Enable 1 to set of Acceptance Filters and fully Acceptance Filtered Mask register, Otherwise the spirit phenomenon will meet a weird one).

- Enable CAN receive interrupt (bit_0 of CANIE (04h) is set to "1").
- Enable CAN transceiver, normal speed, cancel listening only mode (CANMODE (00h) write C8h).
- Set the operating mode to RESET (CANMODE(00h) bit_0 (RSTM) set to 0).
- Enable MCU INT0\ (CS8972 Pin_20 to INT0\).

9.2. Receive CAN Messages

- MCU INT0\ generates an interrupt (receives the INT\ interrupt pin signal in『CS8972』and enters the subroutine interrupt.
- Read 10h to 1Dh register contents in CS8972. (Then CAN receives FIFO in CAN package).
- Open the buffer register (bit_0 of CANCMD (01h) is set to "1").
- Depending on the main requirements of the program.
- Check whether the CAN receive packet count register (CANRXFCNT (7Ah)) is "0"? If the value of this register is not continued reads (10h to 1Dh) thus "0".
- Read 74h and 75h (CS8972 clear the two registers)
- Subroutine interrupt exit.

9.3. Send CAN messages

- Take as an example the 0th group shift register (there are 8 groups).
- Planning 81h (CAN 2.0A use b7=0 or CAN 2.0B use b7=1 and DATA number (0~8)).
- Assume CAN ID ranging from 82h to 85h (no message will be sent if the CAN ID range exceeds the limit).
- CAN DATA from 86h to 8Dh.
- Send the CAN message above using the CAN ID in bit_0 as 05h of its value 1.

10. Trouble Shooting

The CAN protocol is a modified version of Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) similar to Ethernet. If two nodes are trying to send at the same time, instead of collision avoidance, the ID field of the message will resolve itself, which allows the higher priority message to continue and lower priority message has been postponed.

If you found out the device is not working as it should the first thing to do is to check all the connections. Check for shorts or open circuits. Set multimeter to ohms measuring or to buzzer mode and check every connection, to make it easier. If everything on a device seems to work fine, try to find what may cause a problem. This is an obvious step, but sometimes taking a day off, it is way easier to find mistakes (like VCC connected to both sides of an LED), To find if there are any errors in CS8972, test your PCB. If there is a microcontroller, write a sketch to test all the functions of your PCB. Check all of them at once or one at a time. If everything works fine in CS8972 don't worry be happy but if something is wrong, keep reading.