# STAND-ALONE CAN CONTROLLER

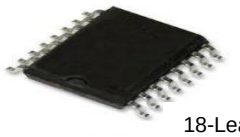
**IN2515**

(Functional equivalent MCP2515 Microchip)

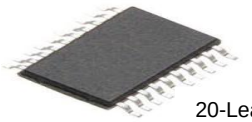
IN2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, version 2.0B

The IC is designed to perform data receiving-transmitting in automotive and industrial applications.

18-Lead PDIP



18-Lead SOIC



20-Lead TSSOP

Fig. 1 Package Types

## FEATURES

- Supply voltage range $V_{CC}$:
  - for IN2515AN, IN2515ADW 2.7 … 5.5 V,
  - for IN2515BN, IN2515BDW; 4.5 … 5.5 V
- Three transmit buffers with prioritizaton and abort features of the transmittance
- Two receive buffers with prioritized message storage
  - Six 29-bit filters
  - Two 29-bit masks
- High-speed SPI Interface (10MHz):
  - SPI modes 0.0 and 1.1
- Implements CAN V2.0B at 1 Mb/s:
  - 0 – 8 byte length in the data field
  - Standard and extended data and remote frames
    - Data byte filtering on the first two data bytes

**ORDERING INFORMATION**

| Device | Operating Temperature Range | Package | Packing |
|---|---|---|---|
| IN2515AN | $T_A$ = -40° to 85° C | DIP-18 | Tube |
| IN2515ADT | | SOP-18 | Tape & Reel |
| IN2515ATSDT | | TSSP-20 | Tube & Reel |
| IN2515BN | $T_A$ = -40° to 125° C | DIP-18 | Tube |
| IN2515BDT | | SOP-18 | Tape & Reel |

## PIN LAYOUT

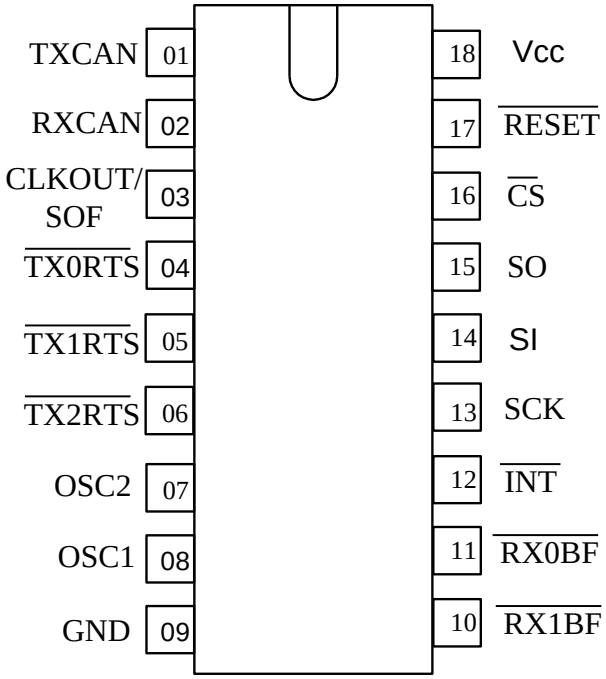



Fig. 2 – IN2515BN, IN2515AN Pin Layout

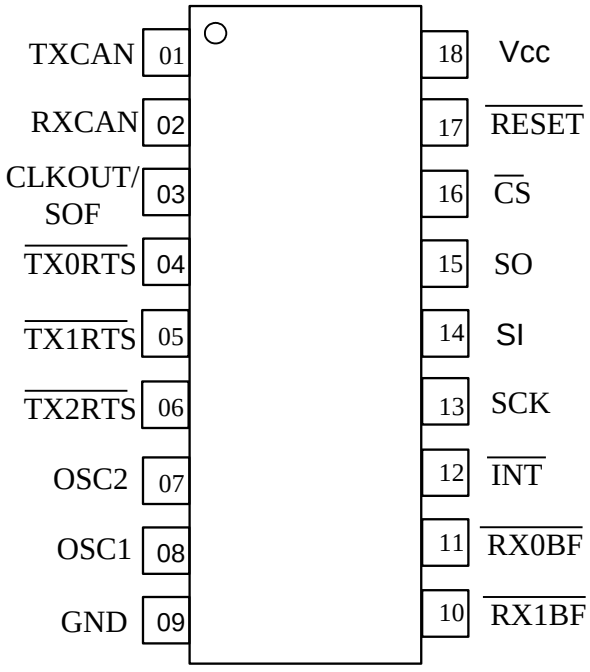



Fig 3 – IN2515BDW, IN2515ADW Pin Layout

Table 1. 18-Lead PDIP / SOIC Package Pin Description

| Pin Number | Symbol | Function |
|---|---|---|
| 01 | TXCAN | Transmit output pin to CAN bus |
| 02 | RXCAN | Receive input pin from CAN bus |
| 03 | CLKOUT / SOF | Clock output pin with programmable prescaler |
| 04 | $\overline{\text{TX0RTS}}$ | Transmit buffer TXB0 request-to-send. 100 kΩ internal pull-up to VDD |
| 05 | $\overline{\text{TX1RTS}}$ | Transmit buffer TXB1 request-to-send. 100 kΩ internal pull-up to VDD |
| 06 | $\overline{\text{TX2RTS}}$ | Transmit buffer TXB2 request-to-send. 100 kΩ internal pull-up to VDD |
| 07 | OSC2 | Oscillator output (Quartz resonator connection) |
| 08 | OSC1 | Oscillator input (Quartz resonator or external clock connection) |
| 09 | GND | Common pin (Ground) |
| 10 | $\overline{\text{RX1BF}}$ | Receive buffer RXB1 interrupt pin or general purpose digital output |
| 11 | $\overline{\text{RX0BF}}$ | Receive buffer RXB0 interrupt pin or general purpose digital output |
| 12 | $\overline{\text{INT}}$ | Interrupt output pin |
| 13 | SCK | SPI Clock input pin for SPI interface |
| 14 | SI | Data input pin for SPI interface |
| 15 | SO | Data output pin for SPI interface |
| 16 | $\overline{\text{CS}}$ | Chip select input pin for SPI interface |
| 17 | $\overline{\text{RESET}}$ | Active low device reset input |
| 18 | $V_{CC}$ | Positive supply for logic and I/O pins |

Notes
1 CAN – Controller Area Network
2 SPI – Serial Peripheral Interface

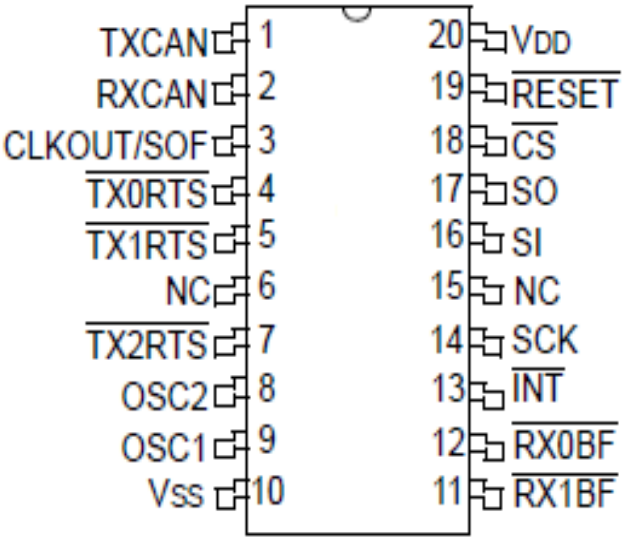



Fig 4 – IN2515ATSD Pin Layout

Table 2. 20-Lead TSSOP Package Pin Description

| Pin Number | Symbol | Function |
|---|---|---|
| 01 | TXCAN | Transmit output pin to CAN bus |
| 02 | RXCAN | Receive input pin from CAN bus |
| 03 | CLKOUT / SOF | Clock output pin with programmable prescaler |
| 04 | $\overline{\text{TX0RTS}}$ | Transmit buffer TXB0 request-to-send. 100 kΩ internal pull-up to VDD |
| 05 | $\overline{\text{TX1RTS}}$ | Transmit buffer TXB1 request-to-send. 100 kΩ internal pull-up to VDD |
| 06 | NC | |
| 07 | $\overline{\text{TX2RTS}}$ | Transmit buffer TXB2 request-to-send. 100 kΩ internal pull-up to VDD |
| 08 | OSC2 | Oscillator output (Quartz resonator connection) |
| 09 | OSC1 | Oscillator input (Quartz resonator or external clock connection) |
| 10 | GND | Common pin (Ground) |
| 11 | $\overline{\text{RX1BF}}$ | Receive buffer RXB1 interrupt pin or general purpose digital output |
| 12 | $\overline{\text{RX0BF}}$ | Receive buffer RXB0 interrupt pin or general purpose digital output |
| 13 | $\overline{\text{INT}}$ | Interrupt output pin |
| 14 | SCK | SPI Clock input pin for SPI interface |
| 15 | NC | |
| 16 | SI | Data input pin for SPI interface |
| 17 | SO | Data output pin for SPI interface |
| 18 | $\overline{\text{CS}}$ | Chip select input pin for SPI interface |
| 19 | $\overline{\text{RESET}}$ | Active low device reset input |
| 20 | $V_{CC}$ | Positive supply for logic and I/O pins |

Notes
1 CAN – Controller Area Network
2 SPI – Serial Peripheral Interface

## BLOCK DIAGRAM

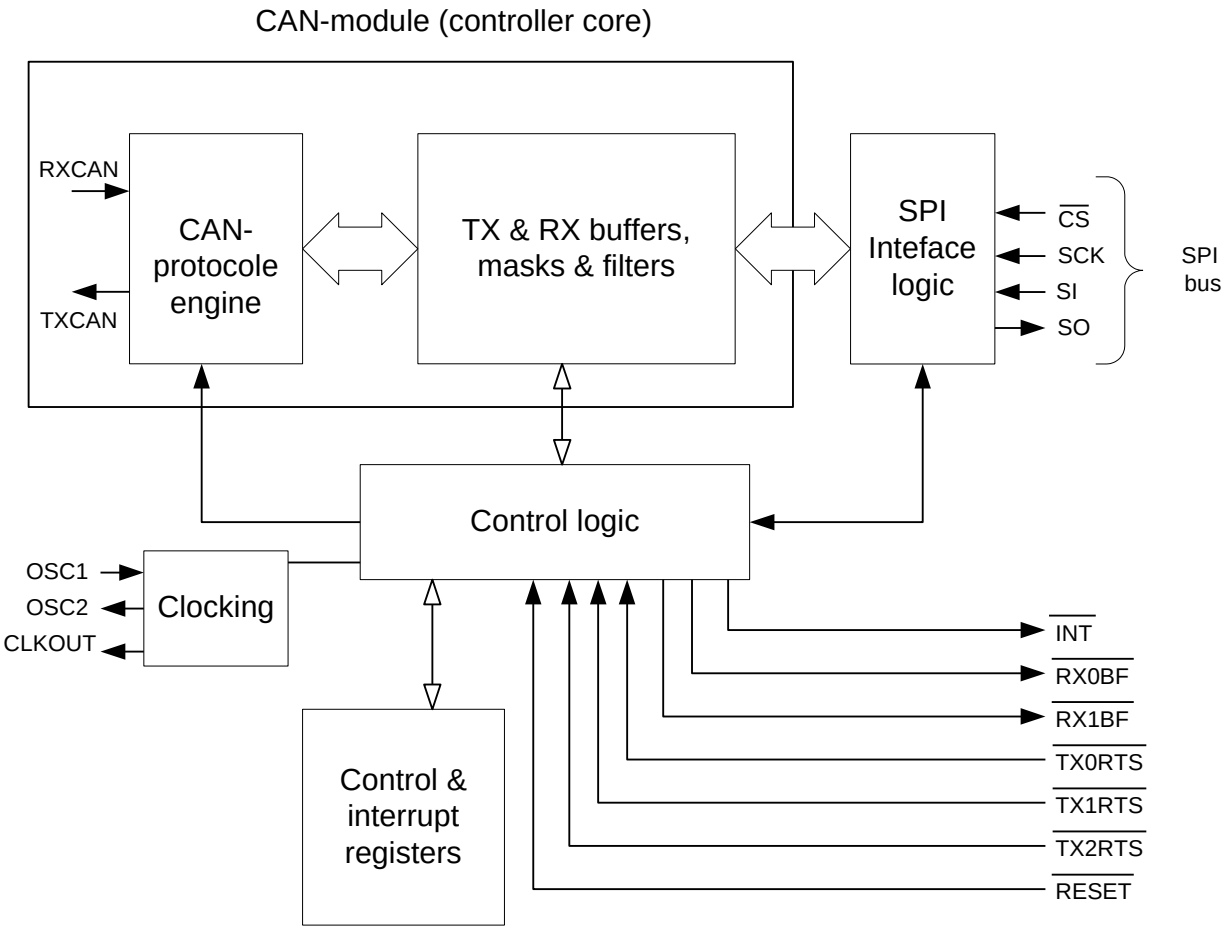



Fig 4 – Block Diagram

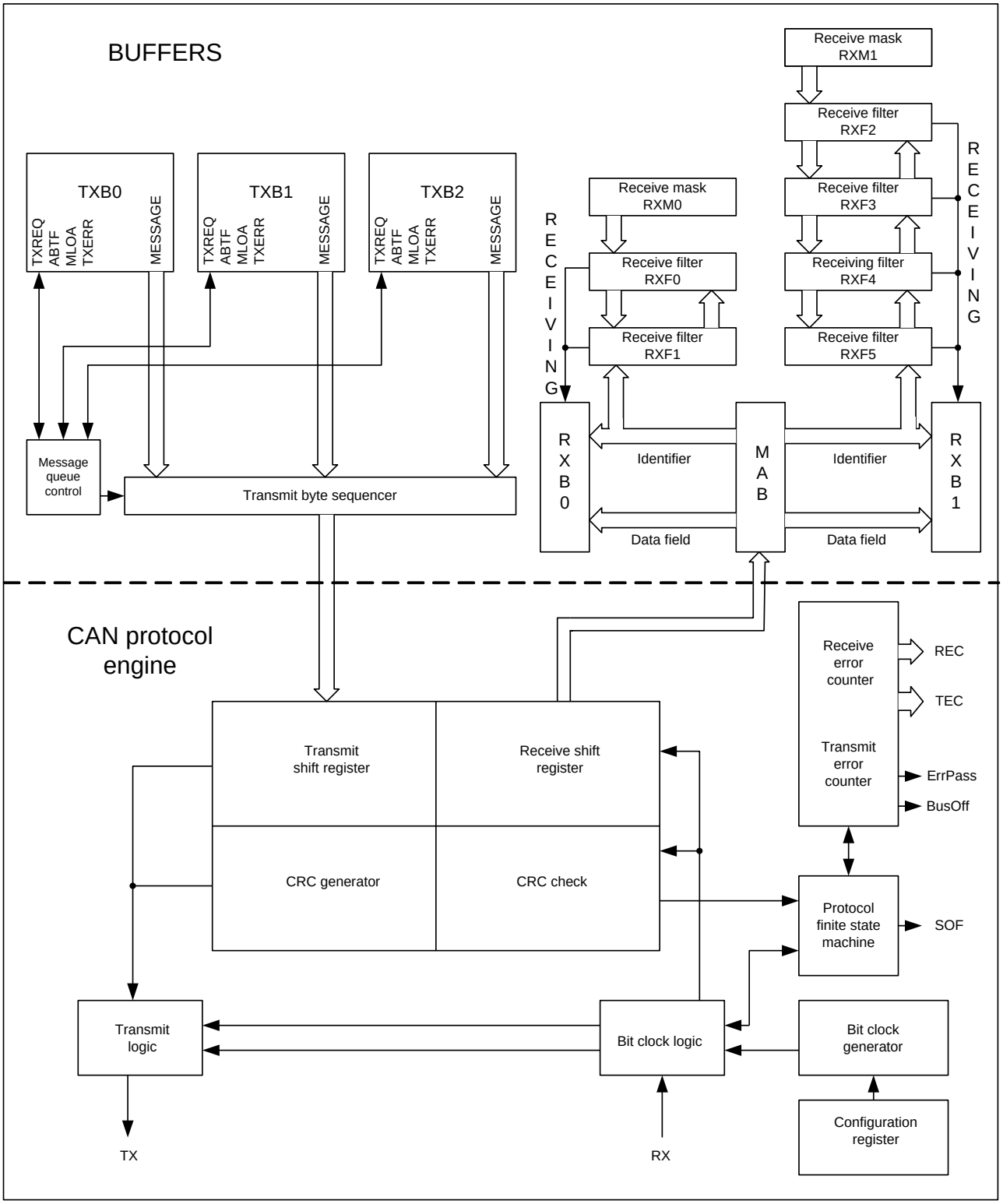



Fig 5 – CAN Module (Controller Core) Block Diagram

## ABSOLUTE MAXIMUM RATING

Table 2

| Symbol | Parameter | Value | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| $V_{CC}$ | Supply voltage<br>- for IN2515AN, IN2515ADW<br>- for IN2515BN, IN2515BDW | <br>-<br>- | <br>7<br>7 | V |
| $V_I$ | Input voltage (all pins) | -0.6 | Vcc+1.0 | V |
| $V_I$ | Input voltage for RXCAN, $\overline{CS}$, $\overline{TXnRTS}$, SCK, SI pins at functional check mode | -0.6 | Vcc+1.0 | V |
| Ta | Ambient (storage) temperature | -60 | 125 | °C |

* Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device.
These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied.
Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

## RECOMMENDED OPERATION MODE

Table 3

| Symbol | Parameter | Value | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| Vcc | Supply voltage<br>- for IN2515AN, IN2515ADW<br>- for IN2515BN, IN2515BDW | <br>2.7<br>4.5 | <br>5.5<br>5.5 | V |
| $V_I$ | Input voltage (all pins) | 0 | Vcc | V |
| $V_I$ | Input voltage for RXCAN, $\overline{CS}$, $\overline{TXnRTS}$, SCK, SI pins at functional check mode | -0.3 | Vcc+1.0 | V |
| Ta | Ambient (operating) temperature<br>- for IN2515AN, IN2515ADW<br>- for IN2515BN, IN2515BDW | <br>-40<br>-40 | <br>85<br>125 | °C |

## DC ELECTRICAL CHARACTERISTICS

Table 4.

| Parameter | Symbol | Value | | Test Conditions | Unit |
|---|---|---|---|---|---|
| | | Min | Max | | |
| Low level output voltage<br>TXCAN pin | $V_{OL1}$ | - | . | $I_{OL}$ = 0.6 mA, $V_{CC}$ = 4.5 V | V |
| $\overline{\text{RXnBF}}$ pin | $V_{OL2}$ | - | 0.6 | $I_{OL}$ = 8.5 mA, $V_{CC}$ = 4.5 V | |
| SO, CLKOUT pins | $V_{OL3}$ | - | 0.6 | $I_{OL}$ = 2.1 mA, $V_{CC}$ = 4.5 V | |
| $\overline{\text{INT}}$ pin | $V_{OL4}$ | - | 0.6 | $I_{OL}$ = 1.6 mA, $V_{CC}$ = 4.5 V | |
| High level output voltage<br>TXCAN pin | $V_{OH1}$ | 3,8 | - | $I_{OH}$ = -3.0 mA, $V_{CC}$ = 4.5 V | V |
| $\overline{\text{RXnBF}}$ pin | $V_{OH2}$ | 3.8 | - | $I_{OH}$ = -3.0 mA, $V_{CC}$ = 4.5V | |
| SO, CLKOUT pins | $V_{OH3}$ | 4.0 | - | $I_{OH}$ = -400 μA,<br>$V_{CC}$ = 4.5 V | |
| $\overline{\text{INT}}$ pin | $V_{OH4}$ | 3.8 | - | $I_{OH}$ = -1.0 mA, $V_{CC}$ = 4.5V | |
| Low level input leakage current<br>RXCAN, SCK, SI, $\overline{\text{CS}}$, $\overline{\text{RESET}}$, CLKOUT, $\overline{\text{RXnBF}}$, SO pins | $I_{ILL1}$ | - | \|-1\| | $V_{IN}$ = 0 V<br>$V_{CC}$ = 5.5 V | μA |
| OSC1 pin | $I_{ILL2}$ | - | \|-5\| | | |
| High level input leakage current<br>RXCAN, SCK, SI, $\overline{\text{CS}}$, $\overline{\text{RESET}}$, CLKOUT, $\overline{\text{RXnBF}}$, SO pins | $I_{ILH1}$ | - | 1 | $V_{CC}$ = 5.5 V<br>$V_{IN}$ = 5.5 V | μA |
| OSC1 pin | $I_{ILH2}$ | - | 5 | | |
| Operating consumption current | $I_{OCC}$ | - | 10 | $V_{CC}$ = 5.5 V<br>$F_{OSC}$ =25 MHz | mA |
| Standby consumption current<br>IN2515ADW, IN2515AN | $I_{CCS}$ | - | 5 | $V_{CC}$ = 5.5 V<br>$V_{IN}$ = 5.5 V | μA |
| IN2515BDW, IN2515BN | | | 8 | | |

*Note*: - for IN2515ADW, IN2515AN Ta from – 40 to + 85 °C,
- for IN2515BDW, IN2515BN Ta from – 40 to + 125 °C

## AC PARAMETERS

Table5.

| Parameter | Symbol | Value | | Test Conditions | Unit |
|---|---|---|---|---|---|
| | | Min | Max | | |
| **CAN- Interface** | | | | | |
| Wake-up noise filter | $T_{WF}$ | 100 | - | $V_{CC}$ = 5.5 V | ns |
| RESET pin low level duration<br>IN2515AN, IN2515ADW | $t_{rl}$ | 2 | - | Vcc = 2.7 V | µs |
| IN2515BN, IN2515BDW | | | | Vcc = 4.5 V | |
| SPI-interface | | | | | |
| Clock frequency | $F_{CLK}$ | - | 10 | - | MHz |
| $\overline{CS}$ setup time | $T_{CSS}$ | 50 | - | - | ns |
| $\overline{CS}$ hold time | $T_{CSH}$ | 50 | - | - | ns |
| $\overline{CS}$ disable time | $T_{CSD}$ | 50 | - | - | ns |
| Data setup time (SI pin) | $T_{SU}$ | 10 | - | - | ns |
| Data hold time (SI pin) | $T_{HD}$ | 10 | - | - | ns |
| Clock high time | $T_{HI}$ | 45 | - | - | ns |
| Clock low time | $T_{LO}$ | 45 | - | - | ns |
| Clock delay time | $T_{CLD}$ | 50 | - | - | ns |
| Clock enable time | $T_{CLE}$ | 50 | - | - | ns |
| Output valid from clock low | $T_V$ | - | 45 | - | ns |
| Output hold time (SO pin) | $T_{HO}$ | 0 | - | - | ns |
| Output disable time (SO pin) | $T_{DIS}$ | - | 100 | - | ns |

*Note*: - for IN2515ADW, IN2515AN Ta from – 40 to + 85 °C,
- for IN2515BDW, IN2515BN Ta from – 40 to + 125 °C

## FUNCTIONALITY

IN2515AN, IN2515ADW**,** IN2515BN**,** IN2515BDW (further IN2515) are stand-alone Controller Area Network (CAN) controllers that implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames. The IN2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead.

The IN2515 is designed to interface directly with the Serial Peripheral Interface (SPI) port available on many microcontrollers and supports Mode 0,0 and Mode 1,1. Commands and data are sent to the device via the SI pin, with data being clocked in on the rising edge of SCK. Data is driven out by the MCP2515 (on the SO line) on the falling edge of SCK. The CS pin must be held low while any operation is performed.

The IN2515 expects the first byte after CS lowered to be the instruction/command byte. This means that CS must be raised and then lowered again to input another command.

Table 6 contains complete list of bytes of SPI instruction set. On detail output and input diagram of both operation modes (Mode 0.0 & Mode 1.1) please refer to Fig 15 & 16.

### SPI Instruction Set

Table 6

| Instruction Name | Instruction Format | Description |
|---|---|---|
| RESET | 1100 0000 | Resets internal registers to default state, set Configuration mode. |
| READ | 0000 0011 | Read data from register beginning at selected address |
| Read RX Buffer | 1001 0nm0 | When reading a receive buffer, reduces the overhead of a normal read command by placing the address pointer at one of four locations, as indicated by ‘n,m’.<br>**Note:** The associated RX flag bit (CANINTF.RXnIF) will be cleared after bringing CS high. |
| WRITE | 0000 0010 | Write data to register beginning at selected address. |
| Load TX Buffer | 0100 0abc | When loading a transmit buffer, reduces the overhead of a normal Write command by placing the address pointer at one of six locations as indicated by ‘a,b,c’. |
| RTS<br><br>(Message Request-To-Send) | 1000 0nnn | Instructs controller to begin message transmission sequence for any of the transmit buffers. |
| Read Status | 1010 0000 | Quick polling command that reads several status bits for transmit and receive functions. |
| RX Status | 1011 0000 | Quick polling command that indicates filter match and message type (standard, extended and/or remote) of received message. |
| Bit Modify | 0000 0101 | Allows the user to set or clear individual bits in a particular register.<br>**Note:** Not all registers can be bit-modified with this command. Executing this command on registers that are not bitmodifiable will force the mask to FFh. |

## RESET

The RESET instruction can be used to re-initialize the internal registers of the IN2515 and set Configuration mode. This command provides the same functionality, via the SPI interface, as the RESET pin. The RESET instruction is a single-byte instruction that requires selecting the device by pulling CS low, sending the instruction byte and then raising CS. It is highly recommended that the reset command be sent (or the RESET pin be lowered) as part of the power-on initialization sequence.

## READ

The READ instruction is started by lowering the CS pin. The READ instruction is then sent to the IN2515 followed by the 8-bit address (A7 through A0). Next, the data stored in the register at the selected address will be shifted out on the SO pin. The internal address pointer is automatically incremented to the next address once each byte of data is shifted out. Therefore, it is possible to read the next consecutive register address by continuing to provide clock pulses. Any number of consecutive register locations can be read sequentially using this method.

The read operation is terminated by raising the CS pin (Figure 6).

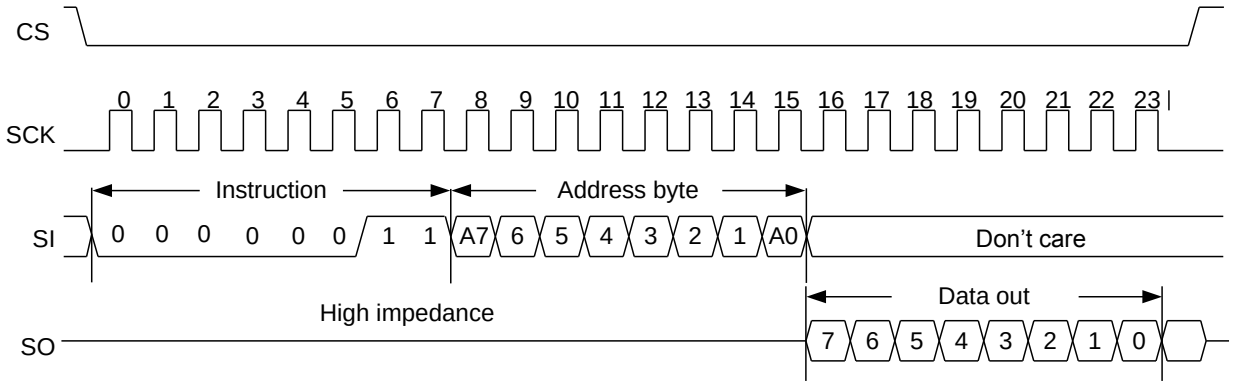
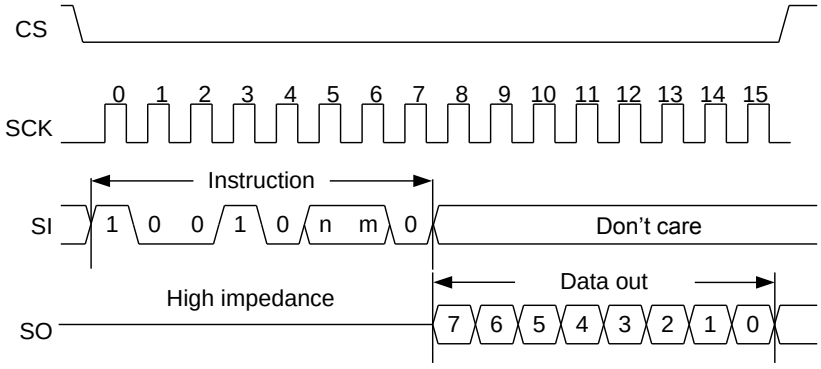



Fig.6 –READ Instruction

## Read RX Buffer

The Read RX Buffer instruction (Figure 7) provides a means to quickly address a receive buffer for reading. This instruction reduces the SPI overhead by one byte, the address byte. The command byte actually has four possible values that determine the address pointer location. Once the command byte is sent, the controller clocks out the data at the address location the same as the READ instruction (i.e., sequential reads are possible). This instruction further reduces the SPI overhead by automatically clearing the associated receive flag (CANINTF.RXnIF) when CS is raised at the end of the command.




| n | m | Address pointer | Address |
|---|---|---|---|
| 0 | 0 | Receive buffer 0, Start at RXB0SIDH | 0x61 |
| 0 | 1 | Receive buffer 0, Start at RXB0D0 | 0x66 |
| 1 | 0 | Receive buffer 1, Start at RXB1SIDH | 0x71 |
| 1 | 1 | Receive buffer 1, Start at RXB1D0 | 0x76 |

Fig. 7 –READ RX BUFFER instruction

## WRITE

The WRITE instruction is started by lowering the CS pin. Then the WRITE instruction is then sent to the IN2515 followed by the address and at least one byte of data. It is possible to write to sequential registers by continuing to clock in data bytes, as long as CS is held low. Data will actually be written to the register on the rising edge of the SCK line for the D0 bit. If the CS line is brought high before eight bits are loaded, the write will be aborted for that data byte and previous bytes in the command will have been written. Refer to the timing diagram in Figure 8 for a more detailed illustration of the byte write sequence.

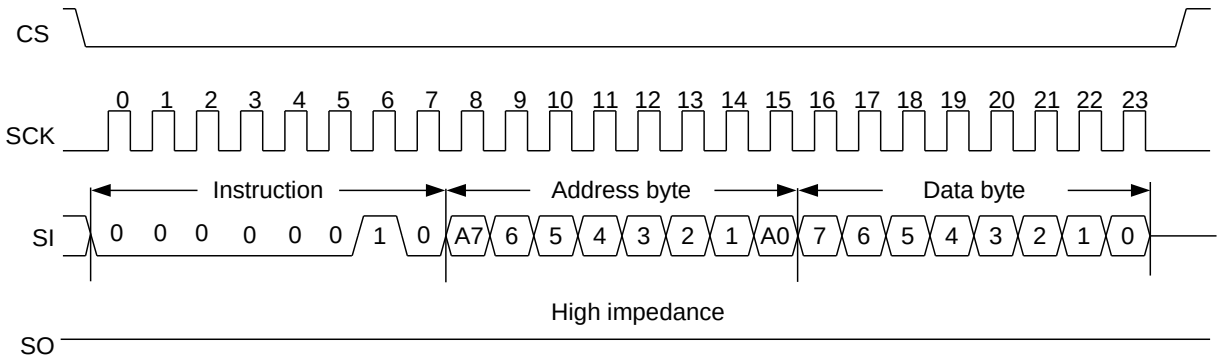
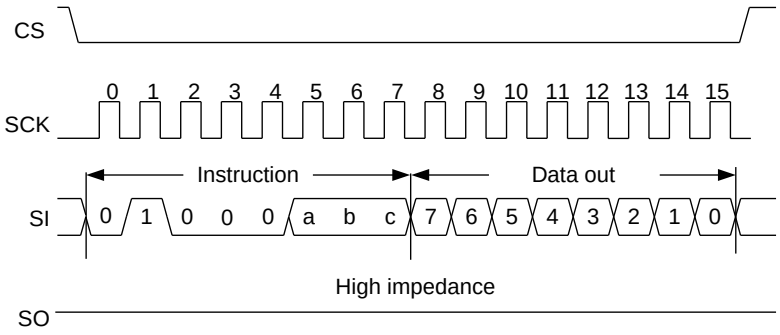



Fig. 8 – BYTE WRITE instruction

## Load TX Buffer

The Load TX Buffer instruction (Figure 9) permits to operate without the eight-bit address required by a normal write command. The eight-bit instruction sets the address pointer to one of six addresses to quickly write to a transmit buffer that points to the “ID” or “data” address of any of the three transmit buffers.




| a | b | c | Address pointer | Address |
|---|---|---|---|---|
| 0 | 0 | 0 | Transmit buffer 0, Start at TXB0SIDH | 0x31 |
| 0 | 0 | 1 | Transmit buffer 0, Start at TXB0D0 | 0x36 |
| 0 | 1 | 0 | Transmit buffer 1, start at TXB1SIDH | 0x41 |
| 0 | 1 | 1 | Transmit buffer 1, Start at TXB1D0 | 0x46 |
| 1 | 0 | 0 | Transmit buffer 2, Start at TXB2SIDH | 0x51 |
| 1 | 0 | 1 | Transmit buffer 2, Start at TXB2D0 | 0x56 |

Fig. 9 – LOAD TX BUFFER instruction

### Request-To-Send (RTS)

The RTS command can be used to initiate message transmission for one or more of the transmit buffers. The IN2515 is selected by lowering the CS pin. The RTS command byte is then sent. Shown in Figure 10, the last 3 bits of this command indicate which transmit buffer(s) are enabled to send. This command will set the TxBnCTRL.TXREQ bit for the respective buffer(s). Any or all of the last three bits can be set in a single command. If the RTS command is sent with nnn = 000, the command will be ignored.

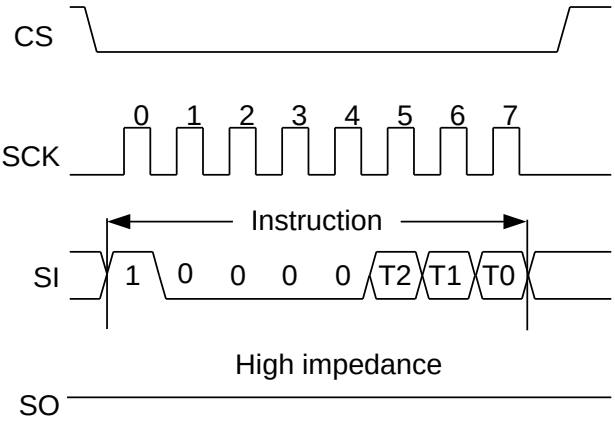



Fig. 10 – REQUEST-TO-SEND instruction (RTS)

### Read Status Instruction

The Read Status instruction allows single instruction access to some of the often used status bits for message reception and transmission. The IN2515 is selected by lowering the CS pin and the read status command byte, shown in Figure 11, is sent to the IN2515. Once the command byte is sent, the IN2515 will return eight bits of data that contain the status.

If additional clocks are sent after the first eight bits are transmitted, the IN2515 will continue to output the status bits as long as the CS pin is held low and clocks are provided on SCK.

Each status bit returned in this command may also be read by using the standard read command with the appropriate register address.

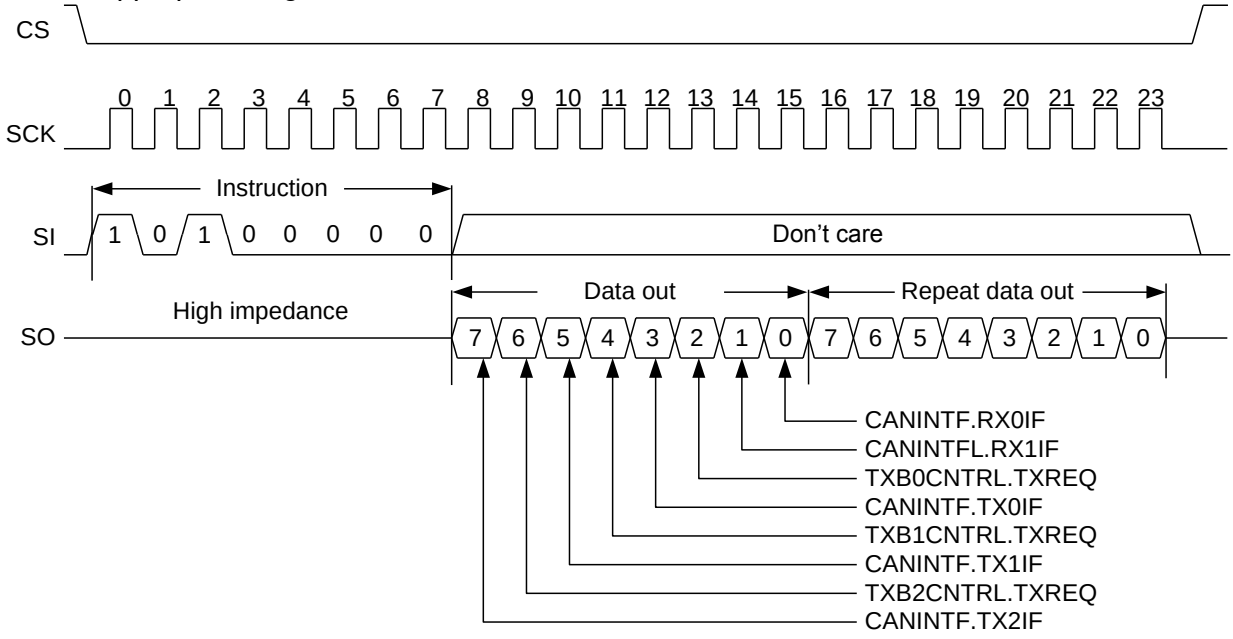
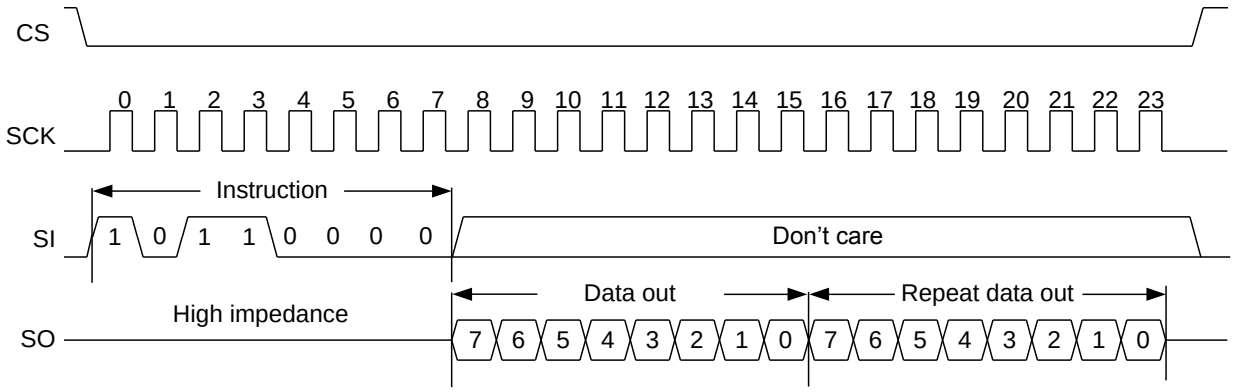



Fig 11 – READ STATUS instruction

## RX Status

The RX Status instruction (Figure 12) is used to quickly determine which filter matched the message and message type (standard, extended, remote). After the command byte is sent, the controller will return 8 bits of data that contain the status data. If more clocks are sent after the 8 bits are transmitted, the controller will continue to output the same status bits as long as the CS pin stays low and clocks are provided.




| 7 | 6 | Received msg |
|---|---|---|
| 0 | 0 | No RX- message |
| 0 | 1 | Message in RXB0 |
| 1 | 0 | Message in RXB1 |
| 1 | 1 | Message in both buffers* |

CANINTF.RXnIF are mapped in bits 7 & 6.

| 4 | 3 | Received msg type. |
|---|---|---|
| 0 | 0 | Standard data frame |
| 0 | 1 | Standard remote frame |
| 1 | 0 | Extended data frame |
| 1 | 1 | Extended remote frame |

Extended ID bit mapped in bit 4.
RTR bit mapped in bit 3 .

| 2 | 1 | 0 | Filter match |
|---|---|---|---|
| 0 | 0 | 0 | RXF0 |
| 0 | 0 | 1 | RXF1 |
| 0 | 1 | 0 | RXF2 |
| 0 | 1 | 1 | RXF3 |
| 1 | 0 | 0 | RXF4 |
| 1 | 0 | 1 | RXF5 |
| 1 | 1 | 0 | RXF0 (rollover to RXB1) |
| 1 | 1 | 1 | RXF1 (rollover to RXB1) |

* Buffer 0 has higher priority, therefore bits, 4:0 display RXB0.status

Fig. 12 – RX STATUS instruction

## Bit Modify Instruction

The Bit Modify instruction provides a means for setting or clearing individual bits in specific status and control registers. This command is available for registers BFPCTRL, TXRTSCTRL, CANCTRL, CNF3, CNF2, CNF1, CANINTE, CANINTF, EFLG, TXB0CTRL, TXB1CTRL, TXB2CTRL, RXB0CTRL, RXB1CTRL only.

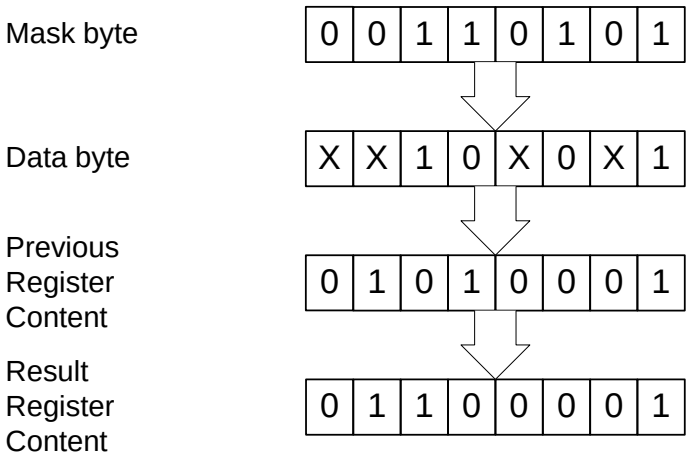



Fig. 13 – Bit Modify instruction

*Note*: Executing the Bit Modify command on registers that are not bit-modifiable will force the mask to FFh. This will allow bytewrites to the registers, not bit modify.

The chip is selected by lowering the CS pin then the Bit Modify command is applied to the IN2515. The command is followed by the address of the register, the mask byte and finally the data byte. The mask byte determines which bits in the register will be allowed to change. A '1' in the mask byte will allow a bit in the register to change, while a '0' will not.

The data byte determines what value the modified bits in the register will be changed to. A '1' in the data byte will set the bit and a '0' will clear the bit, provided that the mask for that bit is set to a '1' (see Figure 14).

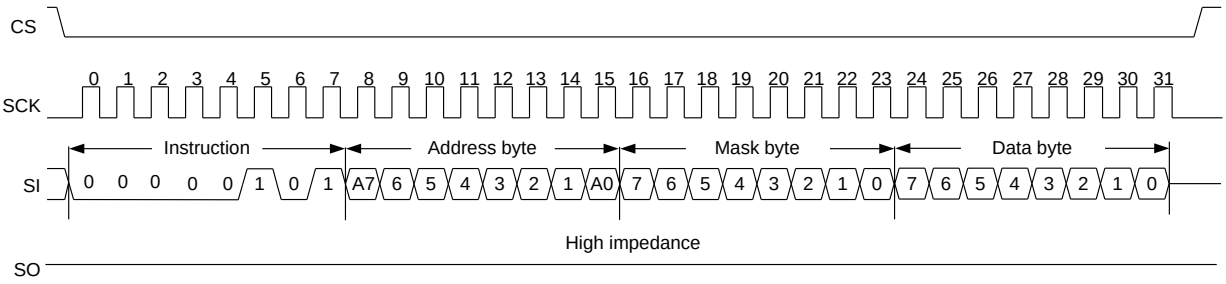



Fig 14 - BIT MODIFY instruction

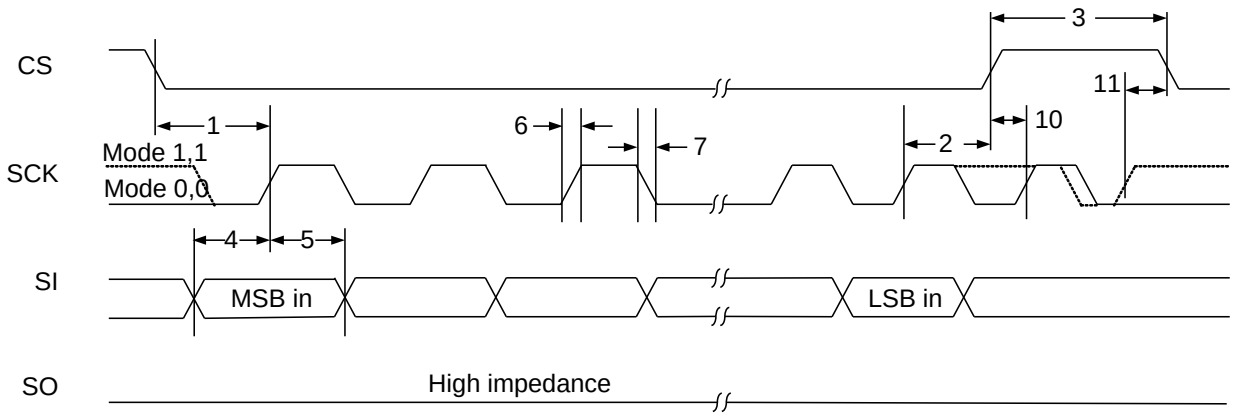



Fig. 15 – SPI- interface input timing

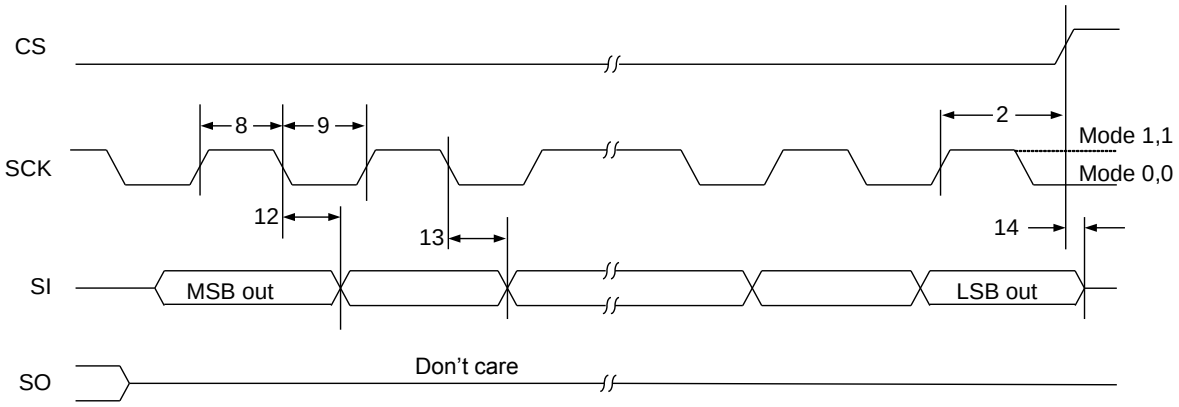



Fig. 16 – SPI- interface output timing

## PACKAGE DIMENSIONS

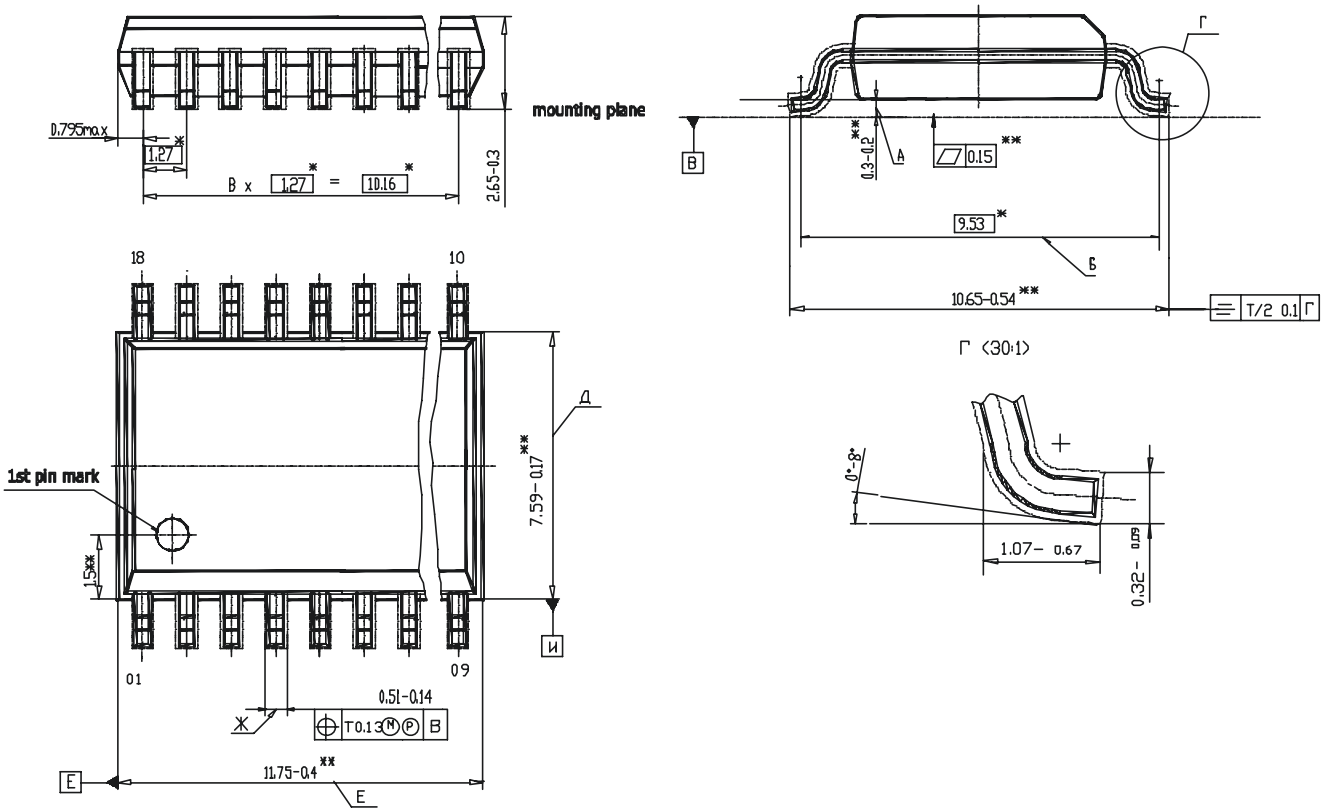



Fig. 17 – **18-Lead Plastic Small Outline (SOIC)**

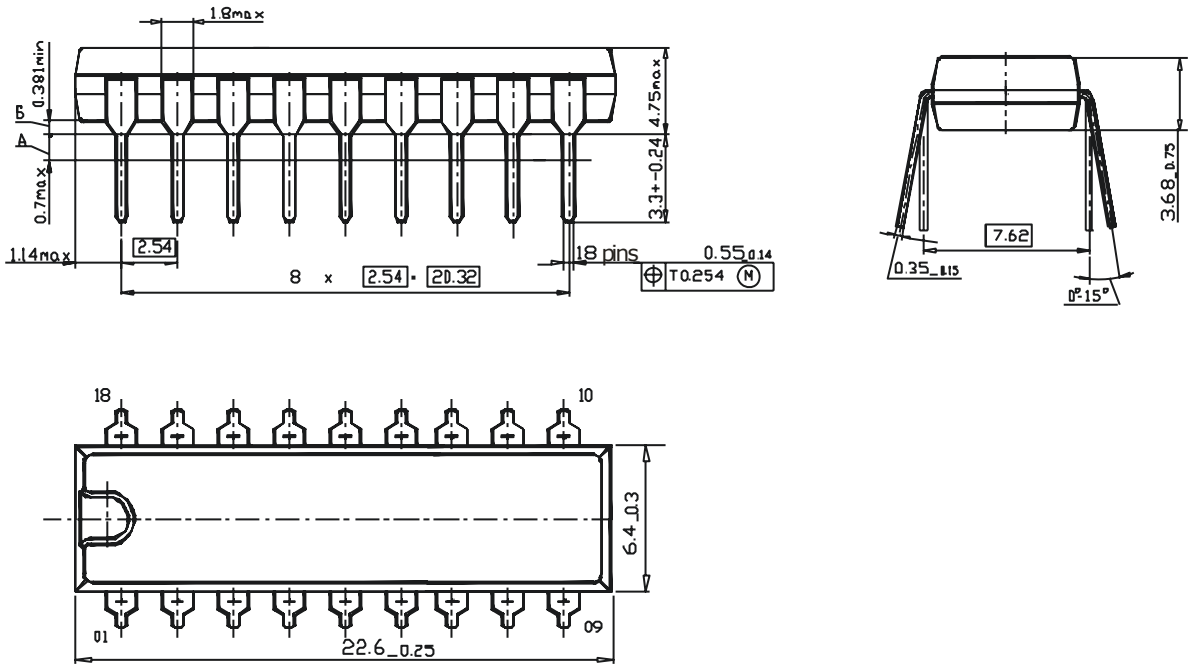
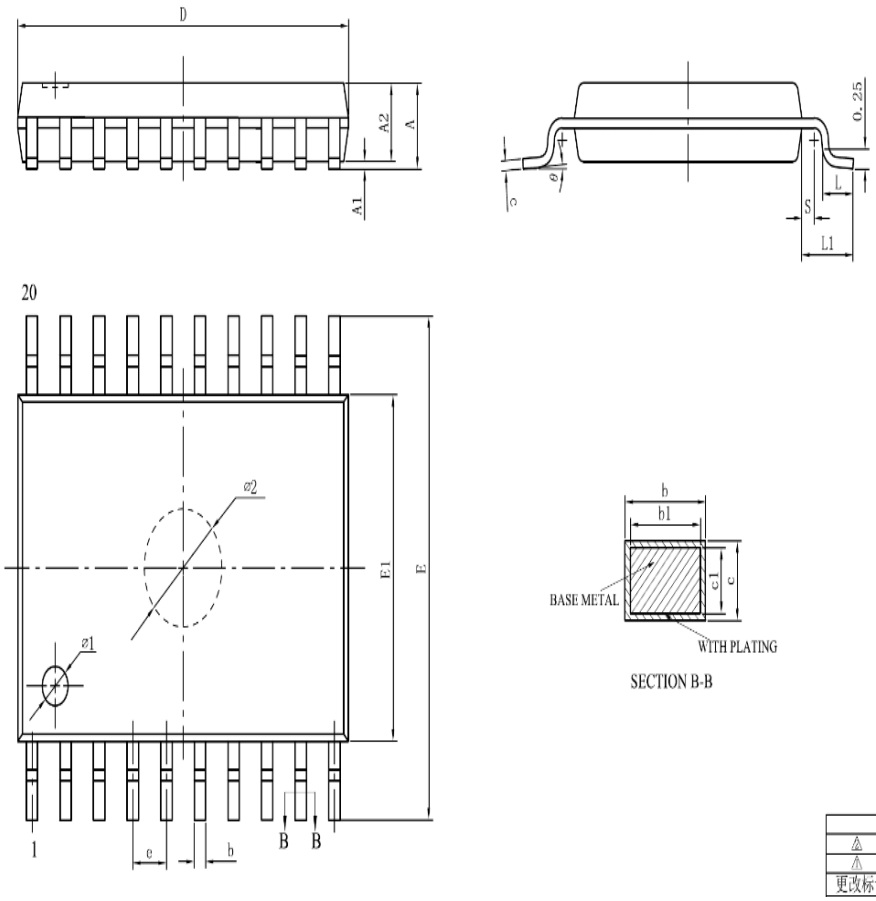



Fig. 18 - **18-Lead Plastic Dual In-line (DIP)**




| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | — | — | 1.20 |
| A1 | 0.05 | — | 0.15 |
| A2 | 0.80 | 1.00 | 1.05 |
| b | 0.19 | — | 0.30 |
| b1 | 0.19 | 0.22 | 0.25 |
| c | 0.09 | — | 0.20 |
| c1 | 0.09 | — | 0.16 |
| D | 6.40 | 6.50 | 6.60 |
| E1 | 4.30 | 4.40 | 4.50 |
| E | 6.20 | 6.40 | 6.60 |
| e | 0.65BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00BSC | | |
| S | 0.20 | — | — |
| Ø1 | Ø0.8X0.05~0.10DP | | |
| Ø2 | Ø1.50X0.05~0.15DP | | |
| θ | 0 | — | 8° |
| L/F载体尺寸 (mil) | 118*165(C) | | |

| | | | |
|---|---|---|---|
| △ | 更改公司名称 | 李方霞 | 2011.7.3 |
| △ | 更改公司名称 | 李翁锋 | 2011.3.3 |
| 更改标记 | 更改内容 | 签 名 | 日 期 |

Fig. 19 - **20-Lead Plastic Thin Shrink Small Outline (TSSOP)**