

# MCP2517FD click

PID: MIKROE-2379

Weight: 33 g

**MCP2517FD click** is a complete CAN solution which carries the MCP2517FD CAN FD controller and ATA6563 high-speed CAN transceiver from Microchip, as well as a DB9 9-pin connector.

The click requires both 3.3V and 5V power supply. It communicates with the target microcontroller through the SPI interface, with additional functionality provided by the following pins on the mikroBUS™ socket: AN, PWM, INT, TX and RX.

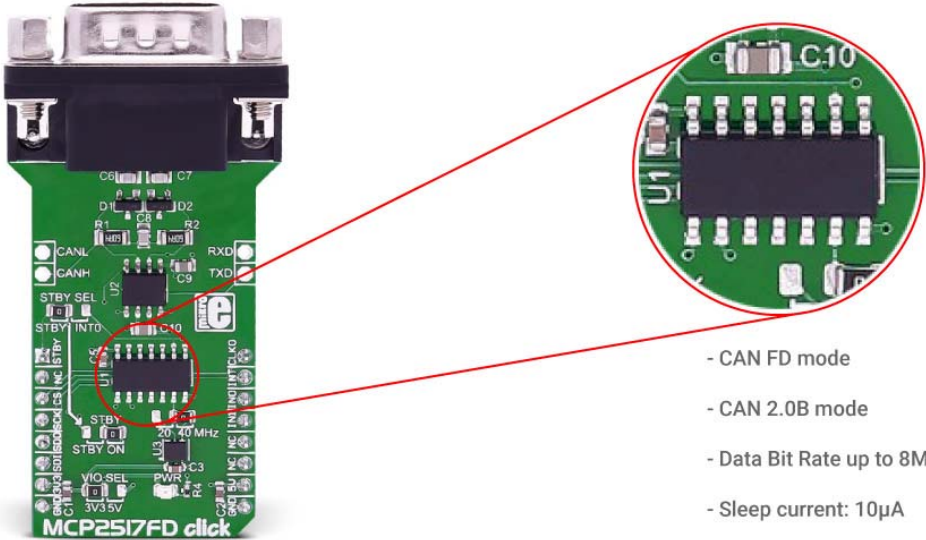
**Note:** For selecting the interface voltage level, use the onboard jumper, and choose between the 3.3V and 5V. For more information, see the Jumpers and Settings table below.



## MCP2517FD features

The MCP2517FD is a cost-effective and small-footprint CAN FD controller that can be easily connected to a microcontroller over an SPI interface. Therefore, a CAN FD channel can be easily added to a microcontroller that is either lacking a CAN FD peripheral, or that doesn't have enough CAN FD channels.

The MCP2517FD supports both, CAN frames in the Classical format (CAN2.0B) and CAN Flexible Data Rate (CAN FD) format.



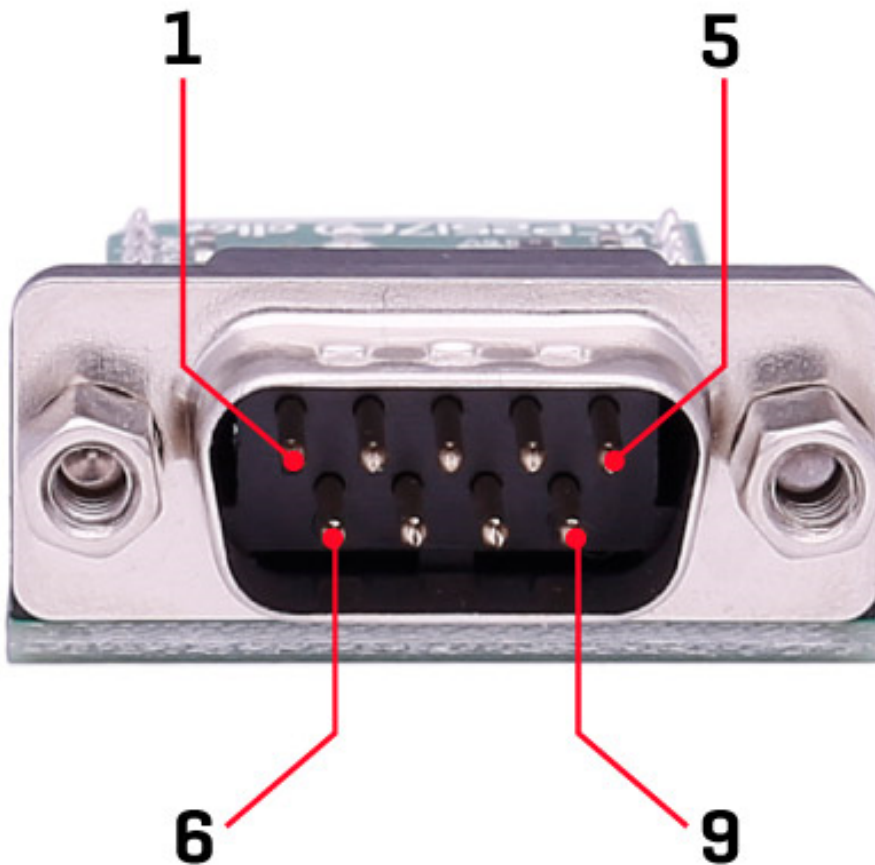
- CAN FD mode
- CAN 2.0B mode
- Data Bit Rate up to 8Mbps
- Sleep current: 10µA

## ATA6563 CAN transceiver features

ATA6563 is a high-speed CAN transceiver that provides an interface between a controller area network (CAN) protocol controller and the physical two-wire CAN bus. The transceiver is designed for high-speed (**up to 5Mbit/s**) CAN applications in the automotive industry, providing differential transmit and receive capability. It offers improved electromagnetic compatibility (EMC) and electrostatic discharge (ESD) performance.

## Connector features

This is a standard DB 9-pin male connector.



## Specifications

<b>Type</b>	CAN
<b>Applications</b>	Simple solution for adding CAN FD connectivity to your device
<b>On-board modules</b>	9-pin CAN connector, ATA6563 CAN transceiver
<b>Key Features</b>	Communication speed up to 5Mbit/s, low electromagnetic emission (EME) and high electromagnetic immunity (EMI)
<b>Interface</b>	SPI
<b>Input Voltage</b>	3.3V or 5V
<b>Compatibility</b>	mikroBUS
<b>Click board size</b>	L (57.15 x 25.4 mm)

## Pinout diagram

This table shows how the pinout on **MCP2517FD click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
Standby mode control input	<b>STBY</b>	1	AN	PWM	16	<b>CLKO</b>	Clock output
	NC	2	RST	INT	15	<b>nINT</b>	Interrupt output

Chip select	<b>nCS</b>	3	CS	TX	14	<b>nINT0</b>	Interrupt output/transceiver standby/GPIO
SPI Clock	<b>SCK</b>	4	SCK	RX	13	<b>nINT1</b>	Interrupt output/GPIO
SPI Master Input Slave Output	<b>MISO</b>	5	MISO	SCL	12	NC	
SPI Master Output Slave Input	<b>MOSI</b>	6	MOSI	SDA	11	NC	
Power supply	<b>+3.3V</b>	7	3.3V	5V	10	<b>+5V</b>	Power supply
Ground	<b>GND</b>	8	GND	GND	9	<b>GND</b>	Ground

## Additional pins

Name	I/O	Description
TX_CAN		CAN transmit
RX_CAN		CAN receive
CANL		CAN low line
CANH		CAN high line

## Jumpers and settings

---

Designator	Name	Default Position	Default Option	Description
JP1	VIO.SEL.	Left	3V3	Power Supply Voltage Selection 3V3/5V, left position 3V3, right position 5V
JP2	STBY	Right	ON	Select Stand by function, default ON, other takes the STBY SEL configuration
JP3	STBY SEL	Left	STBY	Takes STBY input from STBY pin or INTO pin on mikroBUS™
JP4		Right	40MHz	Selects between 20 and 40 MHz clock

## LEDs and buttons

---

Designator	Name	Type	Description
CN1		CONNECTOR	DB9 connector for CAN

## Programming

---

Code examples for MCP2517FD click, written for MikroElektronika hardware and compilers are available on Libstock.

## Code snippet

The following code snippet shows the transmission using MCP2517FD CAN FD controller.

```
01 void transmitMessage( char *msg )
02 {
03     T_MCP2517FD_txFifoEvent    txFlags;
04     T_MCP2517FD_txMsgObj      txObj;
05     uint8_t                    txData[_MCP2517FD_MAX_DATA_BYTES];
06     bool                        flush;
07
08     flush = true;
09     txObj.word[0] = 0;
10     txObj.word[1] = 0;
11
12     txObj.bF.id.SID = 0x300;
13     txObj.bF.id.EID = 0;
14     txObj.bF.ctrl.FDF = 1;
15     txObj.bF.ctrl.BRS = 1;
16     txObj.bF.ctrl.IDE = 0;
17     txObj.bF.ctrl.RTR = 0;
18     txObj.bF.ctrl.DLC = MCP2517FD_DLC_64;
19     txObj.bF.ctrl.SEQ = 1;
20
21     memset(txData, 0, _MCP2517FD_MAX_DATA_BYTES);
22     strcpy(txData, msg);
23     MCP2517FD_TransmitEventGet(MCP2517FD_IDX, &txFlags);
24
25     if( txFlags & MCP2517FD_TX_FIFO_NOT_FULL )
26     {
27         MCP2517FD_TransmitChannelLoad(MCP2517FD_IDX, MCP2517FD_FIFO_CH1, &txObj,
28             txData, MCP2517FD_DlcToDataBytes(txObj.bF.ctrl.DLC), flush);
29         LOG( "\r\n MCP2517FD : Message sent! >" );
30         LOG( txData );
31         LOG( "\r\n" );
32     }
33     else
34     {
35         LOG( "\r\n MCP2517FD : Message not sent! >\r\n" );
36     }
37 }
```