**Listing 1. Pobranie znaku z układu FT22x**

word GetFT22x**()**

**{**

 byte znak**,** st**;**

 SPI\_SSl**();** // SS = L

 FT22xout**(** cmdFT22x\_GetChar **);** // Read request command (CMD=0x01 0x0001)

 **if** **((**st**=**RD\_FT22x**()))** // Odczytaj stan MISO

 **{** // Nie ma znaku

 SPI\_SSh**();** // SS = H

 **return(** NONE **);** // Wyjście z -1 (word = unsigned int)

 **}**

 znak **=** FT22xin**();** // Odczyt linii MIOSIO

 SPI\_SSh**();** // SS = H

 **return(** znak **);**

**}**

**Listing 2. Odczyt znaków z układu FT22x**

FT22xout**(** cmdFT22x\_GetChar **);** // Read request command (CMD=0x01 0x0001)

**while** **(!(**st**=**RD\_FT22x**()))** // Odczytaj stan MISO

**{** // Nie ma znaku

 **\***buf**++** **=** FT22xin**();** // Odczyt linii MIOSIO

**}**

SPI\_SSh**();** // SS = H

**Listing 3. Funkcja przesyłająca ciąg znaków**

byte PrintStringFT22x**(** byte **\***text **)**

**{**

char znak**,** st**;**

 SPI\_SSl**();**

 FT22xout**(**cmdFT22x\_WrChar**);** // Write request command (CMD=0x00)

 st **=** RD\_FT22x**();**

 **if** **(** st **){** SPI\_SSh**();** **return(** false **);** **}**

 **while** **(\***text**)** // Wyswietl tekst z RAMu

 **{**

 znak **=** **(\***text**);**

 FT22xout**(** znak **);**

 text**++;**

 **}**

 SPI\_SSh**();**

 **return(** true **);**

**}**

**Listing 4. Wysłanie bajta do FT201**

byte PutCharFT201**(** byte znak **)**

**{**

 byte st**;**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **return(**false**);**

 //str.27 "AN\_255\_USB to I2C Example using the FT232H and FT201X devices"

 st **=** TWI\_Write**(**FT201WR**);**

 **if** **(**st **!=** 0x18**)**

 **{**

 TWI\_Exit**();**

 **return(**false**);**

 **}** // Brak ACK to bufor nadawczy pełny

 st **=** TWI\_Write**(**znak **);**

 **if** **(**st **!=** 0x28**)**

 **{**

 TWI\_Exit**();**

 **return(**false**);**

 **}**

 //----- STOP -----//

 st **=** TWI\_Stop**();**

 **return(**true**);**

**}**

**Listing 5. Odczytanie znaku z bufora FT201**

word ReadDataFT201**()**

**{**

 byte st**,** dana**=**NONE**;**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **return(**NONE**);**

 st **=** TWI\_Write**(** FT201RD **);**

 // if ( st == 0x48 ){ TWI\_Exit(); return( NONE ); } // Brak ACK oznacza brak znaku do odbioru

 **if** **(**st **!=** 0x40**)**

 **{**

 TWI\_Exit**();**

 **return(**NONE**);**

 **}**

 // Odczytaj ostatni bajt

 dana **=** TWI\_Read**(** false **);**

 **if** **(**TWI\_Status **!=** 0x58**)**

 **{**

 TWI\_Exit**();**

 **return(** NONE **);**

 **}**

 //----- STOP -----//

 st **=** TWI\_Stop**();**

 **return(**dana**);**

**}**

**Listing 6. Odczyt liczby znaków w buforze FIFO**

byte AvailableFT201**()**

**{**

 byte st**,** ava**,** repeat**=**FT201REPEAT**;**

 **do**

 **{**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **continue;**

 // General adres

 st **=** TWI\_Write**(**0**);**

 **if** **(**st **!=** 0x18**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 //str.25 "AN\_255\_USB to I2C Example using the FT232H and FT201X devices"

 //DataAvailable command

 st **=** TWI\_Write**(**0x0C**);**

 **if** **(**st **!=** 0x28**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **return(**0**);**

 st **=** TWI\_Write**(**FT201RD**);**

 **if** **(**st **!=** 0x40**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 // Odczytaj ostatni bajt

 ava **=** TWI\_Read**(**false**);**

 **if** **(**TWI\_Status **!=** 0x58**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 //----- STOP -----//

 st **=** TWI\_Stop**();**

 **return(**ava**);**

 **}** **while(**repeat**--);**

 **return** 0**;**

**}**

**Listing 7. Sprawdzenie statusu układy FT201**

byte ReadStFT201**()**

**{**

 byte st**,** repeat**=**FT201REPEAT**;**

 StFT201 **=** **-**1**;**

 **do**

 **{**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **continue;**

 // General adres

 st **=** TWI\_Write**(**0**);**

 **if** **(**st **!=** 0x18**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 //str.22 "AN\_255\_USB to I2C Example using the FT232H and FT201X devices"

 //cmd status

 st **=** TWI\_Write**(**0x16**);**

 **if** **(**st **!=** 0x28**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **continue;**

 st **=** TWI\_Write**(**FT201RD**);**

 **if** **(**st **!=** 0x40**)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

 // Odczytaj ostatni bajt

 StFT201 **=** TWI\_Read**(** false **);**

 **if** **(**TWI\_Status **!=** 0x58 **)**

 **{**

 TWI\_Exit**();**

 **continue;**

 **}**

//0x00 Suspended

//0x01 Default

//0x02 Addressed

//0x03 Configured

 //----- STOP -----//

 st **=** TWI\_Stop**();**

 **}** **while(** repeat**--** **);**

 **return(** StFT201 **);**

**}**

**Listing 8. Odczytanie identyfikatora układu FT201**

byte ReadIdFT201**()**

**{**

 byte st**;**

 IdFT201**[**0**]** **=** IdFT201**[**1**]** **=** IdFT201**[**2**]** **=** **-**1**;**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **return;**

 //str.22 "AN\_255\_USB to I2C Example using the FT232H and FT201X devices"

 //cmd read ID

 st **=** TWI\_Write**(** 0xF8/\*0x7C\*/ **);**

 **if** **(**st **!=** 0x18**)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

 st **=** TWI\_Write**(**FT201RD**);**

 **if** **(**st **!=** 0x28**)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

 //----- START -----//

 st **=** TWI\_Start**();**

 **if** **(**st **!=** 8 **&&** st **!=** 0x10**)** **return(**st**);**

 //ID command 0x7C

 st **=** TWI\_Write**(**0xF9/\*0x7c\*/**);**

 **if** **(**st **!=** 0x40**)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

 IdFT201**[**0**]** **=** TWI\_Read**(**true**);**

 **if** **(**TWI\_Status **!=** 0x50 **)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

 IdFT201**[**1**]** **=** TWI\_Read**(**true**);**

 **if** **(**TWI\_Status **!=** 0x50**)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

 // Odczytaj ostatni bajt

 IdFT201**[**2**]** **=** TWI\_Read**(**false**);**

 **if** **(**TWI\_Status **!=** 0x58**)**

 **{**

 TWI\_Exit**();**

 **return;**

 **}**

//ID[0] - pewnie vid pid jeśli różny od domyślnego

//ID[1] -

//ID[2] - wartośc ustawiana w FT\_PROG

 //----- STOP -----//

 st **=** TWI\_Stop**();**

**}**

**Listing 9. Ustawienie bitu odpowiedzialnego za ładowanie VCP**

ReadMtpFT22x**(** 0**,** 256**,** mtp **);**

crc **=** CalculateMtpCrc**(** **(**word**\*)**mtp **);**

crcL**=**crc**;**

**if** **((**crcL**==**mtp**[**0xFE**])** **&&** **(**crc**>>**8**==**mtp**[**0xFF**]))**

**{**

 //jeśli CRC ok to zapis

 WriteMtpFT22x**(** 0x24**,** data **);**

 //

 mtp**[**0**]** **|=** 0x80**;** // Ustaw ładowanie VCP

 crc **=** CalculateMtpCrc**(** **(**word**\*)**mtp **);**

 mtp**[**0xFE**]** **=** crc**;** mtp**[**0xFF**]** **=** crc **>>** 8**;**

 WriteMtpFT22x**(** 0**,** mtp**[**0**]** **);**

 WriteMtpFT22x**(** 0xFE**,** mtp**[**0xFE**]** **);**

 WriteMtpFT22x**(** 0xFF**,** mtp**[**0xFF**]** **);**

**}**