Sprawozdanie Asynchroniczna transmisja szeregowa realizowana przez interfejs USART/RS232

Autorzy: Radosław Niżnik, Adrian Żerebiec

Zadanie 3.2

Za pomocą kodu przedstawionego poniżej oraz całego osprzętowania potrzebnego do zadania zmierzyliśmy potrzebne wartości z oscylogramu.
Kod programu:

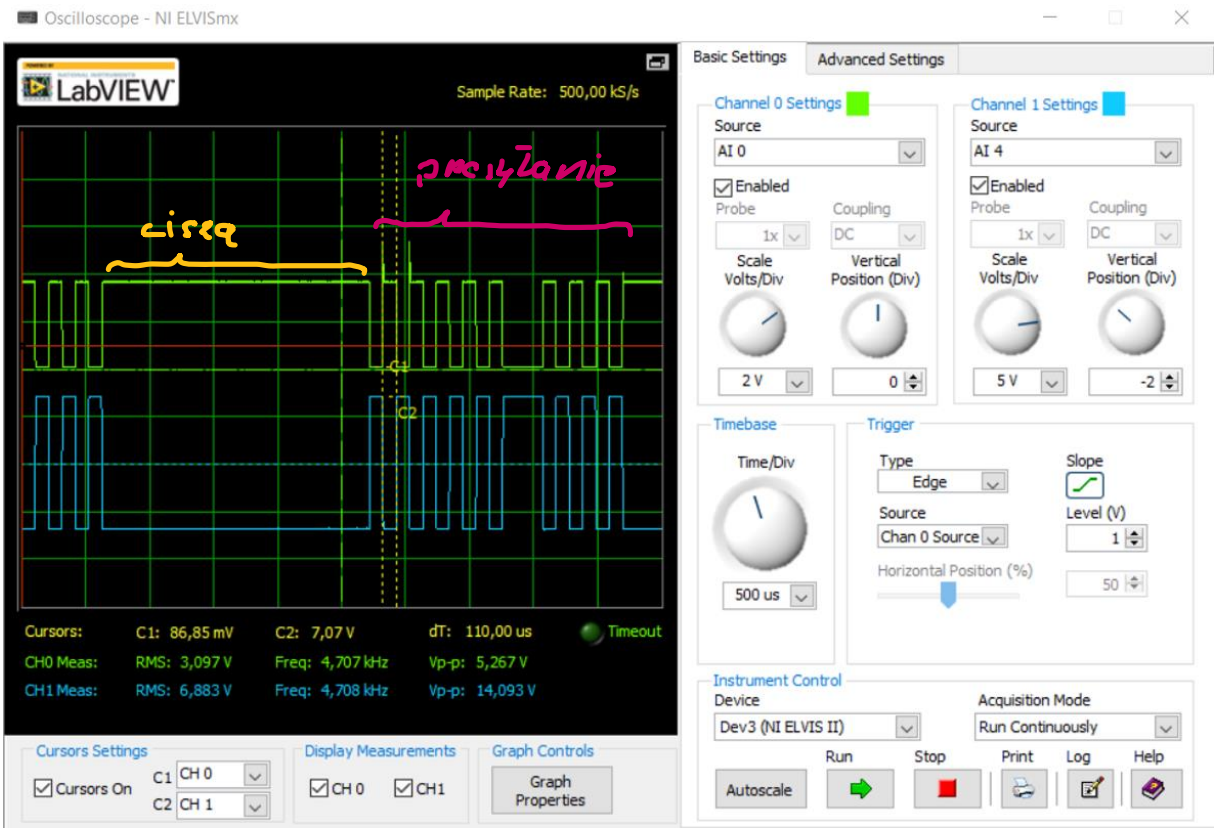
```
void setup() {
  //Initialize serial and wait for port to open:

  Serial1.begin(9600);
  while (!Serial1) {
    ; // wait for serial port to connect. Needed for
  }
}


void loop() {

  Serial1.write("U");
  delay(3);
  Serial1.write("U");
  Serial1.write("T");
  delay(10);

}
```

Wybraliśmy do testu znak „U”, gdyż wydawał nam się on najlepszy do wykonywania pomiarów ze względu na swoją postać binarną: ‘01010101’. „T” natomiast w zapisie binarnym wygląda następująco: ‘01010100’. Naszym zdaniem łatwo będzie obserwować na tych znakach zmiany napięć.

Wartości odczytane z oscylogramu przebiegów sygnałów obserwowanych na wyjściach USART_TX i RS232_TX:

	Wyjście USART_TX	Wyjście RS232_TX
Napięcie odpowiadające bitowi o wartości **1**	3,68V	-6,84V
Napięcie odpowiadające bitowi o wartości **0**	86,85mV	7,11V
Prędkość transmisji, format ramki	9600	
Czas przesłania całej ramki (SDU)	1,04ms	
Czas trwania pojedynczego bitu o wart. **0**	100us	
Czas trwania pojedynczego bitu o wart. **1**	110us	

Zrzut ekranu z oscyloskopem:



Wnioski

Bardzo łatwo możemy transmitować dane z pomocą write na wyjścia USART_TX oraz RS232_TX. Jednak bardzo ważne jest, aby ustawić dobrą częstotliwość odbierania nadawania i odbierania. Warto zauważyć, że dla wyjścia RS232_TX w momencie, gdy mamy wartość 1 napięcie jest ujemne, natomiast dla 0 jest ono dodatnie. Zatem łatwo stwierdzić, że USART_TX to górna cześć wyników (ta na zielono), a na niebiesko mamy RS232_TX. Dodatkowo widzimy, że czas przesłania całej ramki to 1,04ms, a przesłanie pojedynczego bitu zajmuje około 105 us. Zmierzony przez nas czas przesyłania pojedynczej ramki zgadza się z przypuszczeniami, gdyż szybkość przesyłania u nas ustawiona była na 9600, więc jest to bardzo zbliżony wynik.

Zadanie 3.3

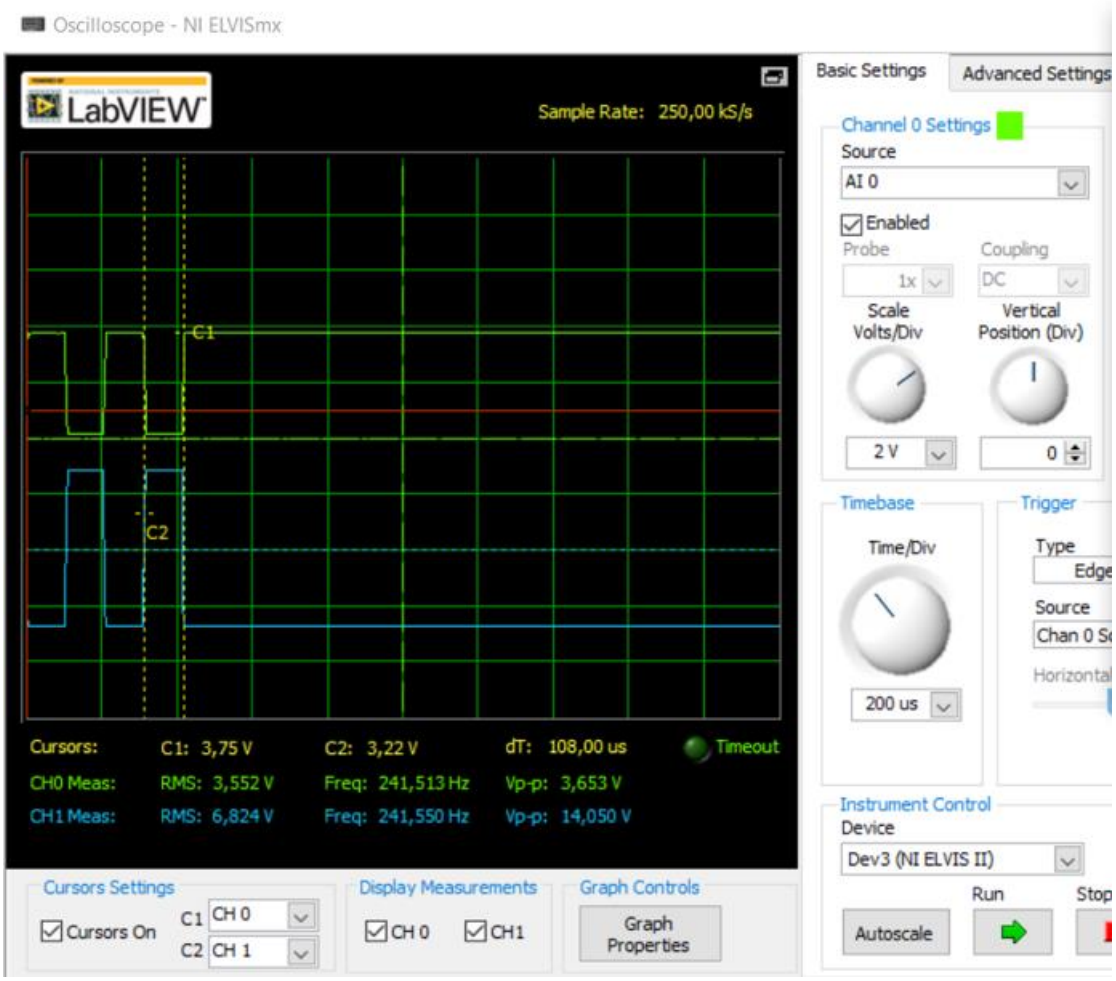
W celu napisania zdania musieliśmy skorzystać z dokumentacji ATmega2560. W ten sposób z delikatnymi usprawnieniami udało nam się napisać kod.

Kod zadania:

```
#define FOSC 16000000
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1

void initsd(void){
  USART_Init( MYUBRR);
}

void USART_Init( unsigned int ubrr){
  UBRR1H = (unsigned char)(ubrr>>8);
  UBRR1L = (unsigned char)ubrr;
  UCSR1B = (1<<RXEN1)|(1<<TXEN1);
  UCSR1C = (1<<USBS1)|(3<<UCSZ10);
}

void USART_Transmit( unsigned int data )
{
  while ( !( UCSR1A & (1<<UDRE1)));
  /* Copy 9th bit to TXB8 */
  UCSR1B &= ~(1<<TXB81);
  if ( data & 0x0100 )
  UCSR1B |= (1<<TXB81);
  /* Put data into buffer, sends the data */
  UDR1 = data;
}


void setup() {
  initsd();
}


void loop() {
  USART_Transmit(0x55);
  delay(5);
}
```

Ponownie jako znak, który chcemy zaobserwować wybieramy „U” czyli 0x55.

Zaobserwowane wyniki na oscyloskopie:



Sprawdzenie, czy w programie Putty są wyniki: