Embedded System
Assignment 7

Submitted By:

Pratyush Jaiswal
18EE30021

AIM:

Proteus implementation of data communication using USART (port D of the Atmega32). Transmit the letters “I I T Kharagpur” from one microcontroller to another microcontroller and display it. Data can be transmitted using the ASCII code.

Atmel Code for Transmitting

```
;
; Transmit_code.asm
;
; Created: 14-03-2021 01:55:06
; Author : Pratyush Jaiswal
;

; Replace with your application code
.ORG 0x0
      JMP INIT
.ORG UDREaddr
      JMP ISR_Transmit


INIT:
      LDI R16,HIGH(RAMEND)
      ; initialize high
      OUT SPH,R16
            ; byte of SP
      LDI R16,LOW(RAMEND)
            ; initialize low
      OUT SPL,R16
            ; byte of SP

.EQU F_CPU=1000000
      ; frequency is 1MHz
.EQU USART_BAUDRATE=4800                                                ;
set baud rate to for serial comm
.EQU BAUD_PRESCALE = (((F_CPU/(USART_BAUDRATE*16)))-1)  ; calculate the
scaling factor obtained from formula in dataset

USART_Init:
      LDI R16, 0
```

```
      LDI R16, HIGH(BAUD_PRESCALE)                                    ;storing
the HIGHER 8 bits of prescaler
      STS UBRR0H, R16
      LDI R16, LOW(BAUD_PRESCALE)
      ;storing the LOWER 8 bits of prescaler
      STS UBRR0L, R16

      ; Enable receiver and transmitter
      LDI r16, (1<<RXEN0) | (1<<TXEN0)
      STS UCSR0B, R16

      ; Set frame format: 8data
      LDI r16, (1<<UCSZ01)|(1<<UCSZ00)                                ;enabling
8 bit frame of data
      STS UCSR0C, R16


MAIN:
      LDI R16, 'I'
      RCALL USART_TRANSMIT
      LDI R16, ' '
      RCALL USART_TRANSMIT
      LDI R16, 'I'
      RCALL USART_TRANSMIT
      LDI R16, ' '
      RCALL USART_TRANSMIT
      LDI R16, 'T'
      RCALL USART_TRANSMIT
      LDI R16, ' '
      RCALL USART_TRANSMIT
      LDI R16, ' '
      RCALL USART_TRANSMIT
      LDI R16, 'K'
      RCALL USART_TRANSMIT
      LDI R16, 'h'
      RCALL USART_TRANSMIT
      LDI R16, 'a'
      RCALL USART_TRANSMIT
      LDI R16, 'r'
      RCALL USART_TRANSMIT
      LDI R16, 'a'
      RCALL USART_TRANSMIT
      LDI R16, 'g'
      RCALL USART_TRANSMIT
      LDI R16, 'p'
      RCALL USART_TRANSMIT
      LDI R16, 'u'
      RCALL USART_TRANSMIT
      LDI R16, 'r'
      RCALL USART_TRANSMIT
      LDI R16, ' '
      RCALL USART_TRANSMIT
      LDI R16, 13
      RCALL USART_TRANSMIT
```

```
      LDI R16, 10
      RCALL USART_TRANSMIT
      RJMP MAIN
      RJMP EXIT

USART_TRANSMIT:
            ; Wait for empty transmit buffer
            LDS R20, UCSR0A
            SBRS R20, UDRE0
            RJMP USART_Transmit
            ; Put data (r16) into buffer, sends the data
            sts UDR0, R16
            RET

ISR_transmit:
      STS UDR0,R16
      RETI
exit:
      RJMP exit
```

Atmel Code for Receiving

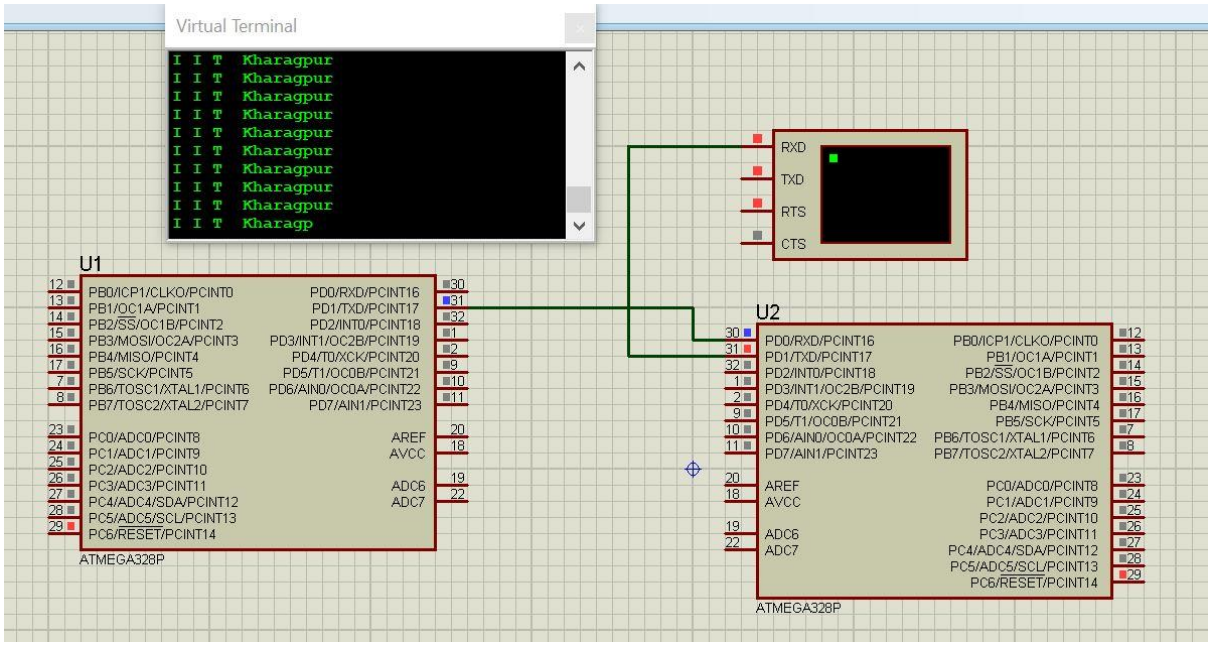
```
;
; Receiver_code.asm
;
; Created: 16-03-2021 01:57:29
; Author : Pratyush Jaiswal
;
; Replace with your application code

.EQU F_CPU=1000000
      ; frequency is 1MHz
.EQU USART_BAUDRATE=4800                                                ;
set baud rate to for serial comm
.EQU BAUD_PRESCALE = (((F_CPU/(USART_BAUDRATE*16)))-1)  ; calculate the
scaling factor obtained from formula in dataset

LDI R16,HIGH(RAMEND)                                                    ;
initialize high
OUT SPH,R16
      ; byte of SP
LDI R16,LOW(RAMEND)
      ; initialize low
OUT SPL,R16
      ; byte of SP


USART_Init:
      LDI R16, 0
      LDI R16, HIGH(BAUD_PRESCALE)                                ;storing
the HIGHER 8 bits of prescaler
      STS UBRR0H, R16
      LDI R16, LOW(BAUD_PRESCALE)
      ;storing the LOWER 8 bits of prescaler
      STS UBRR0L, R16
```

```
      ; Enable receiver and transmitter
      LDI r16, (1<<RXEN0) | (1<<TXEN0)
      STS UCSR0B, R16

      ; Set frame format: 8data
      LDI r16, (1<<UCSZ01)|(1<<UCSZ00)                              ;enabling
8 bit frame of data
      STS UCSR0C, R16



MAIN:
      RCALL USART_RECEIVE
      ; receiving from the transmitter
      RCALL USART_TRANSMIT                                                ;
after receiving, transmit the received data to the screen
      RJMP MAIN

USART_RECEIVE:
            ; Wait for data to be received
            LDS R20, UCSR0A
            SBRS R20, RXC0
            RJMP USART_RECEIVE
            ; Get and return received data from buffer
            LDS R16, UDR0
            RET

USART_TRANSMIT:
            ; Wait for empty transmit buffer
            LDS R20, UCSR0A
            SBRS R20, UDRE0
            RJMP USART_Transmit
            ; Put data (r16) into buffer, sends the data
            sts UDR0, R16
            RET

exit:
            RJMP exit
```

Proteus Schematic and Demo Result

Setting:

Baud rate of the virtual terminal: 4800


Virtual Terminal
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagpur
I I T Kharagp
RXD
TXD
RTS
CTS
U1
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
AREF
AVCC
ADC6
ADC7
ATMEGA328P
U2
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
AREF
AVCC
ADC6
ADC7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
ATMEGA328P