EMBEDDED SYSTEMS CMPE-453

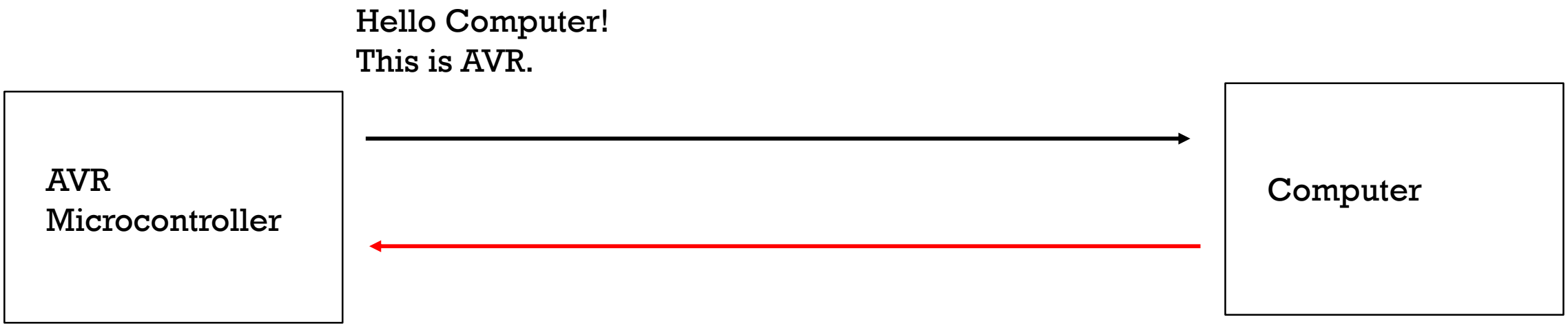
Department of Computer Engineering

Serial Communication-3

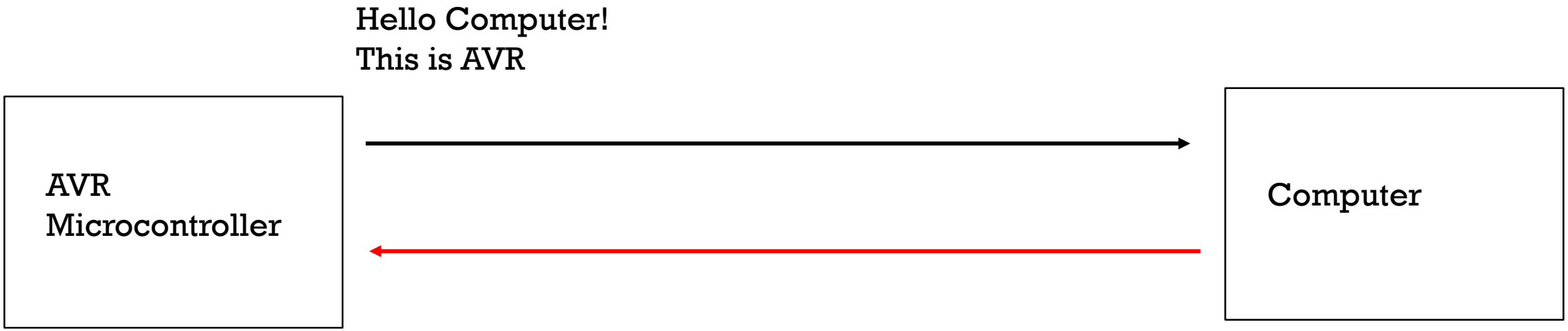
TRANSMITTING A STRING OVER UART

```
void printString(const char myString[])
      {
      uint8_t i = 0;
/* Srings are ended by a null-charachetr in C */
      while (myString[i])
            {
            transmitByte(myString[i]);
            i++;
            }
      }
```

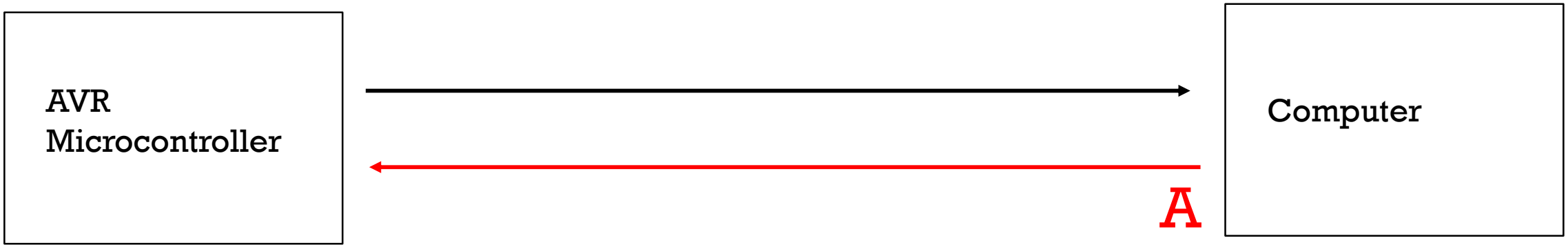
AVR TALKING TO COMPUTER



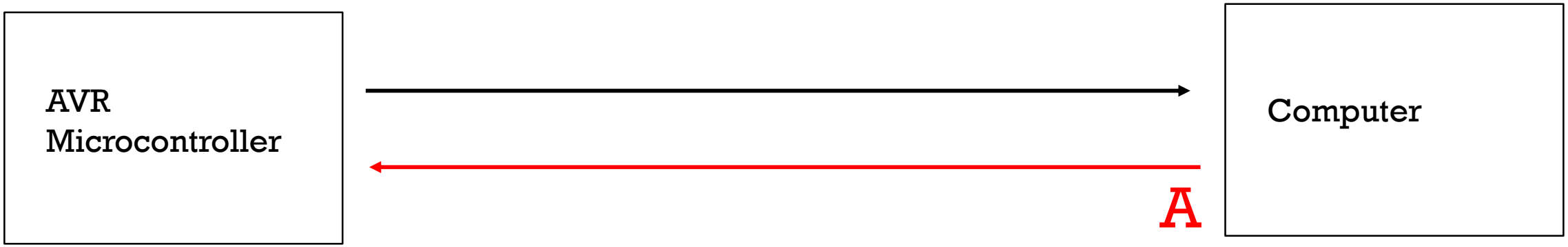
AVR TALKING TO COMPUTER



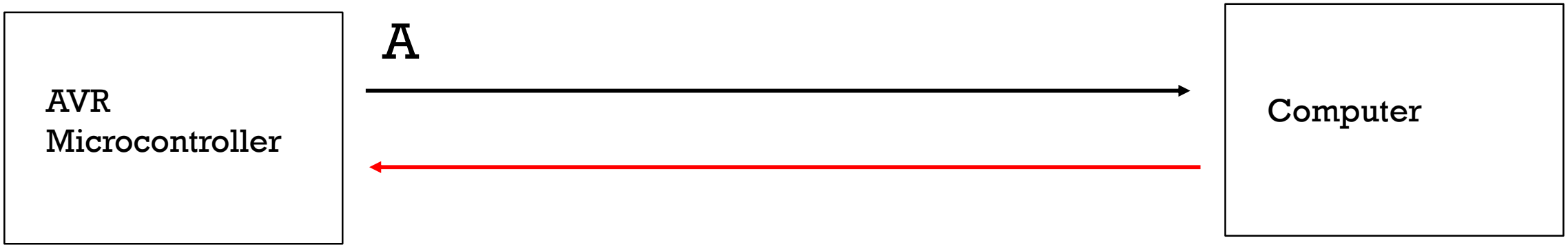
AVR TALKING TO COMPUTER



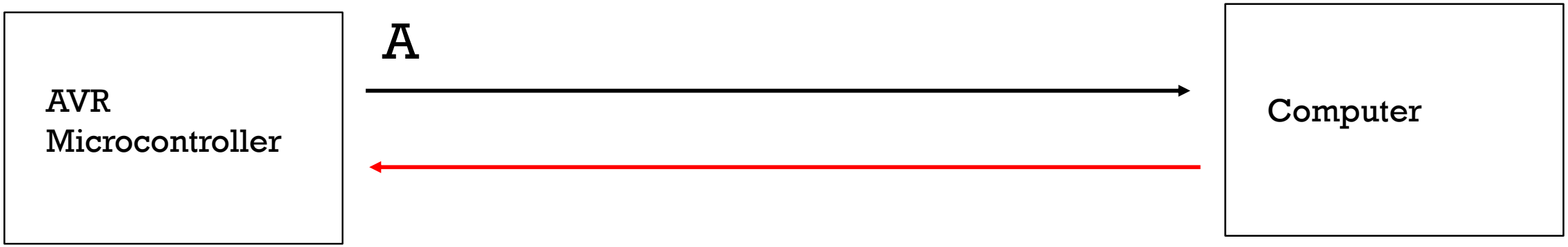
AVR TALKING TO COMPUTER



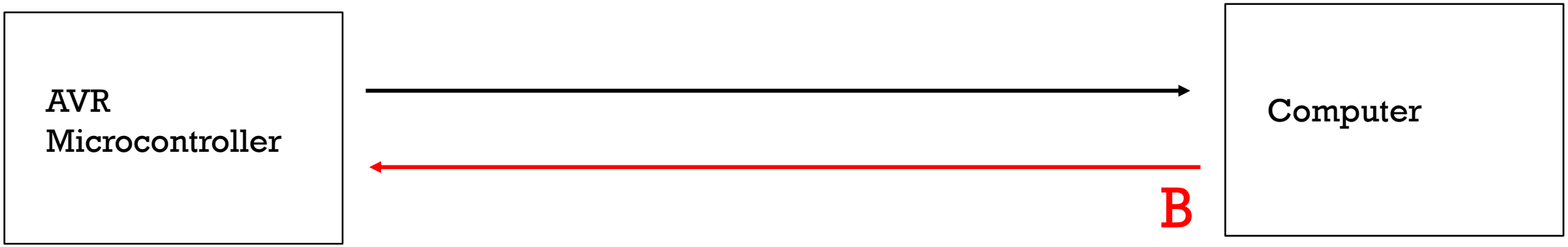
AVR TALKING TO COMPUTER



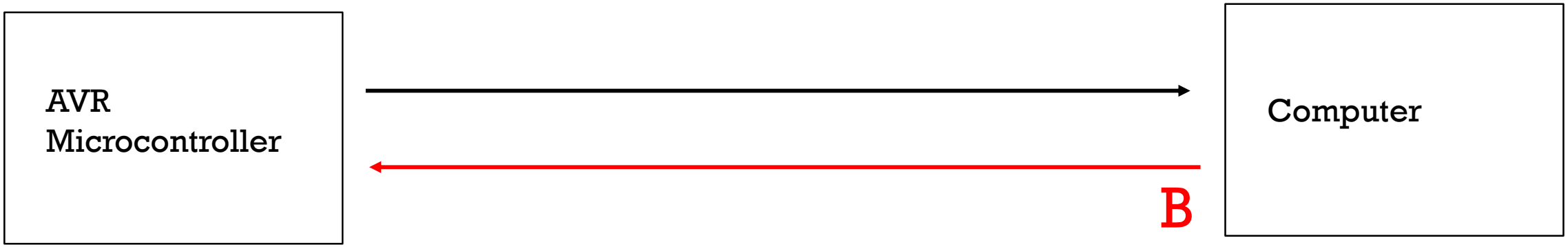
AVR TALKING TO COMPUTER



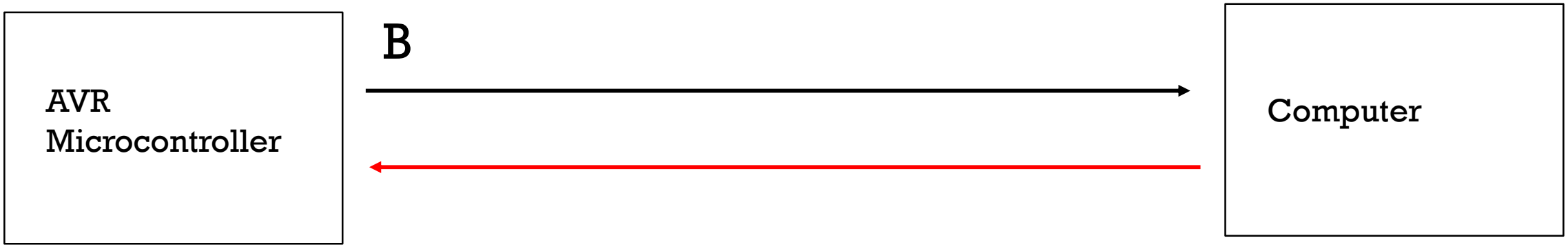
AVR TALKING TO COMPUTER



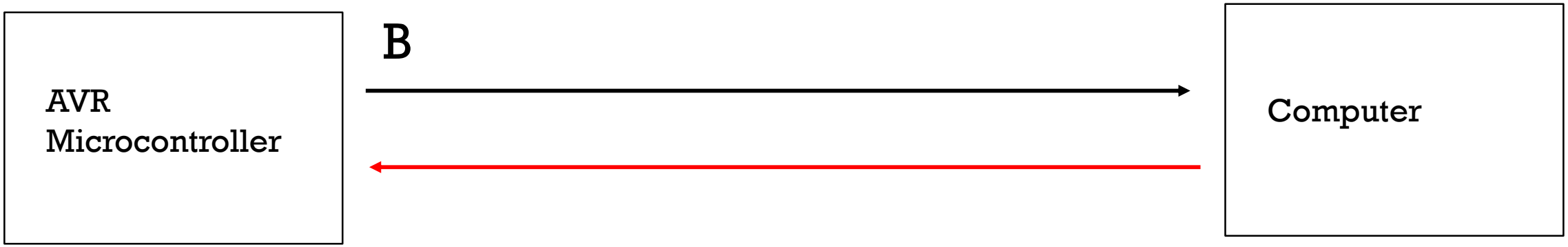
AVR TALKING TO COMPUTER



AVR TALKING TO COMPUTER



AVR TALKING TO LAPTOP



EXAMPLE LOOPBACK COMMUNICATION

Develop an AVR based circuit that

1. Sends a greetings message to computer
2. Receives a byte of data from computer
3. Send the same received byte back to computer
4. Go to step 2

PSEUDOCODE

```
Initialize UART

Send Greetings to computer

while(1)
        {
        receive byte from computer
        send same byte back to computer
        }
```

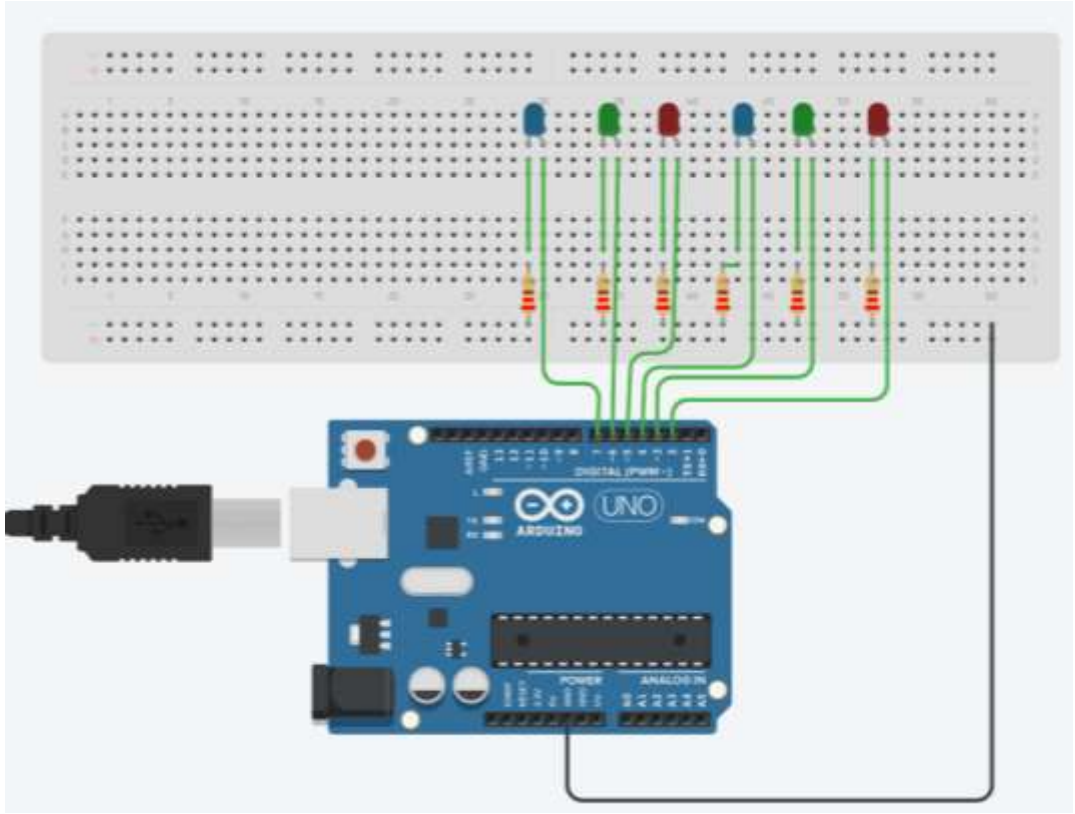
C-PROGRAM

```
// ------- Preamble -------- //
#include <binaryMacro.h>
#include <macros.h>
#include <pinDefines.h>
#include <portpins.h>
#include <USART.c>

#include <avr/io.h>
#include <util/delay.h>
```

```
int main(void)
        {
        char serialCharacter;
        // ----- Inits --------- //
        initUSART();
        /* to test */
        printString("Hello computer!\r\n");
        printString("This is AVR!\r\n");
        // ------ Event loop ------ //
        while (1)
                {
                serialCharacter = receiveByte();
                transmitByte(serialCharacter);

                }
        return 0;
}
```

EXAMPLE

- Build an AVR circuit and program it such that
  - Through UART, it asks a laptop to send an index value.
  - Reads the index value
  - Turns on index-th LED out of 6 LEDs connected to the output pins (as shown in the figure).
  - If index is less than 2 or greater than 7, AVR shall send an «out of range» message to laptop



PSEUDOCODE

```
Initialize UART
Initialize output port
Send Greetings to Laptop

while(1)
        {
        Send prompt to laptop
        Receive index value
        If value is not in range
                Send «Out of range» message
        else
                Turn on index-th LED
        }
```

```
#include <binaryMacro.h>
#include <macros.h>
#include <pinDefines.h>
#include <portpins.h>
#include <USART.c>
// ------- Preamble -------- //
/* A program that turns on an LED at index value received from UART receiver. */
#include <avr/io.h>
#include <util/delay.h>

int main(void)
      {
      char serialCharacter;
      // -------- Inits --------- //
      LED_DDR = 0xff; /* set up LEDs for output */
      DDRD = 0xff;
      initUSART();
       /* to test */
      printString("Hello Laptop!\r\n");
      printString("This is AVR.\r\n");
      // ------ Event loop ------ //
      while (1)
            {
            printString("Enter a number, from 2 to 7 and I will turn on corresponding LED\r\n");
            serialCharacter = receiveByte();
            transmitByte(serialCharacter);
            printString("\n");
            if ((serialCharacter < 0x32) || (serialCharacter > 0x37))
                  printString("The number you entered is out of range \r\n");
            else
                  {
                  serialCharacter = serialCharacter-0x30;

                  LED_PORT= (1 << serialCharacter);

            PORTD = LED_PORT;
                  }
            }/* End event loop */
      return 0;
      }
```