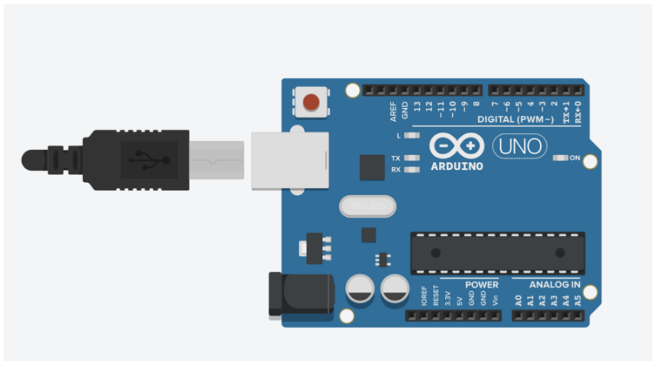
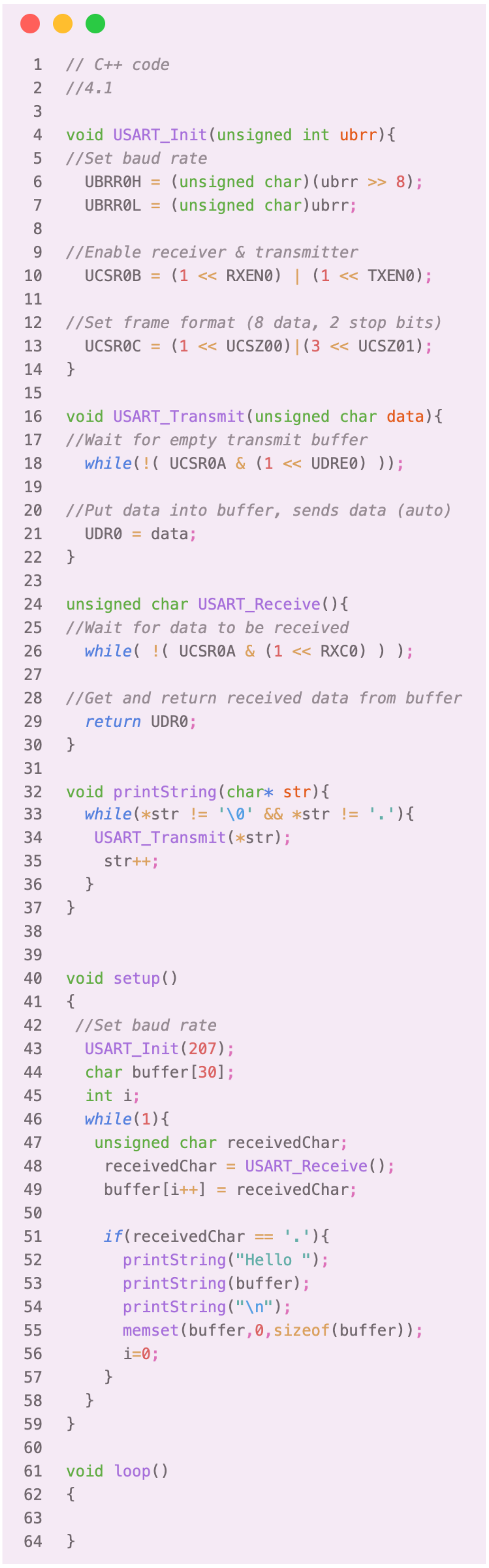
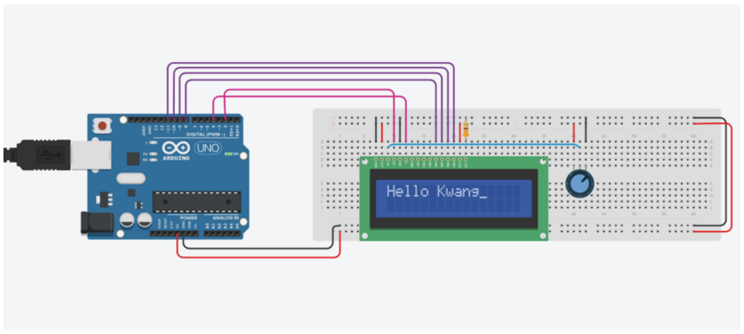
Lab4 Problem1: AVR USART Programming

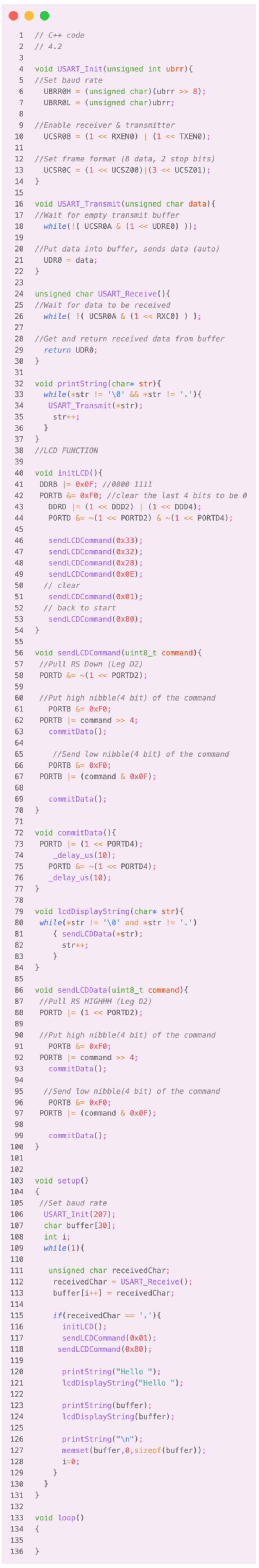




```
// C++ code
//4.1

void USART_Init(unsigned int ubrr){
//Set baud rate
  UBRR0H = (unsigned char)(ubrr >> 8);
  UBRR0L = (unsigned char)ubrr;

//Enable receiver & transmitter
  UCSR0B = (1 << RXEN0) | (1 << TXEN0);

//Set frame format (8 data, 2 stop bits)
  UCSR0C = (1 << UCSZ00)|(3 << UCSZ01);
}

void USART_Transmit(unsigned char data){
//Wait for empty transmit buffer
  while(!( UCSR0A & (1 << UDRE0) ));

//Put data into buffer, sends data (auto)
  UDR0 = data;
}

unsigned char USART_Receive(){
//Wait for data to be received
  while( !( UCSR0A & (1 << RXC0) ) );

//Get and return received data from buffer
  return UDR0;
}

void printString(char* str){
  while(*str != '\0' && *str != '.'){
   USART_Transmit(*str);
    str++;
  }
}


void setup()
{
 //Set baud rate
  USART_Init(207);
  char buffer[30];
  int i;
  while(1){
   unsigned char receivedChar;
    receivedChar = USART_Receive();
    buffer[i++] = receivedChar;

    if(receivedChar == '.'){
      printString("Hello ");
      printString(buffer);
      printString("\n");
      memset(buffer,0,sizeof(buffer));
      i=0;
    }
  }
}

void loop()
{

}
```

Lab4 Problem2: AVR USART Programming




```cpp
// C++ code
// 4.2

void USART_Init(unsigned int ubrr){
//Set baud rate
  UBRR0H = (unsigned char)(ubrr >> 8);
  UBRR0L = (unsigned char)ubrr;

//Enable receiver & transmitter
  UCSR0B = (1 << RXEN0) | (1 << TXEN0);

//Set frame format (8 data, 2 stop bits)
  UCSR0C = (1 << UCSZ00)|(3 << UCSZ01);
}

void USART_Transmit(unsigned char data){
//Wait for empty transmit buffer
  while(!( UCSR0A & (1 << UDRE0) ));

//Put data into buffer, sends data (auto)
  UDR0 = data;
}

unsigned char USART_Receive(){
//Wait for data to be received
  while( !( UCSR0A & (1 << RXC0) ) );

//Get and return received data from buffer
  return UDR0;
}

void printString(char* str){
  while(*str != '\0' && *str != '.'){
   USART_Transmit(*str);
    str++;
  }
}
//LCD FUNCTION

void initLCD(){
 DDRB |= 0x0F; //0000 1111
 PORTB &= 0xF0; //clear the last 4 bits to be 0
   DDRD |= (1 << DDD2) | (1 << DDD4);
   PORTD &= ~(1 << PORTD2) & ~(1 << PORTD4);

   sendLCDCommand(0x33);
   sendLCDCommand(0x32);
   sendLCDCommand(0x28);
   sendLCDCommand(0x0E);
  // clear
   sendLCDCommand(0x01);
  // back to start
   sendLCDCommand(0x80);
}

void sendLCDCommand(uint8_t command){
 //Pull RS Down (Leg D2)
 PORTD &= ~(1 << PORTD2);

 //Put high nibble(4 bit) of the command
   PORTB &= 0xF0;
 PORTB |= command >> 4;
   commitData();

    //Send low nibble(4 bit) of the command
   PORTB &= 0xF0;
 PORTB |= (command & 0x0F);

   commitData();
}

void commitData(){
 PORTD |= (1 << PORTD4);
    _delay_us(10);
   PORTD &= ~(1 << PORTD4);
   _delay_us(10);
}

void lcdDisplayString(char* str){
 while(*str != '\0' and *str != '.')
    { sendLCDData(*str);
      str++;
    }
}

void sendLCDData(uint8_t command){
 //Pull RS HIGHHH (Leg D2)
 PORTD |= (1 << PORTD2);

 //Put high nibble(4 bit) of the command
   PORTB &= 0xF0;
 PORTB |= command >> 4;
   commitData();

  //Send low nibble(4 bit) of the command
   PORTB &= 0xF0;
 PORTB |= (command & 0x0F);

   commitData();
}


void setup()
{
 //Set baud rate
  USART_Init(207);
  char buffer[30];
  int i;
  while(1){

   unsigned char receivedChar;
    receivedChar = USART_Receive();
    buffer[i++] = receivedChar;

    if(receivedChar == '.'){
      initLCD();
      sendLCDCommand(0x01);
     sendLCDCommand(0x80);

      printString("Hello ");
      lcdDisplayString("Hello ");

      printString(buffer);
      lcdDisplayString(buffer);

      printString("\n");
      memset(buffer,0,sizeof(buffer));
      i=0;
    }
  }
}

void loop()
{

}
```