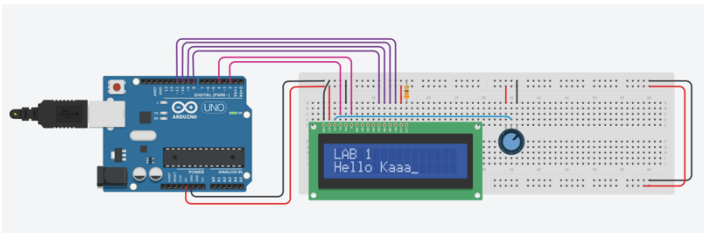
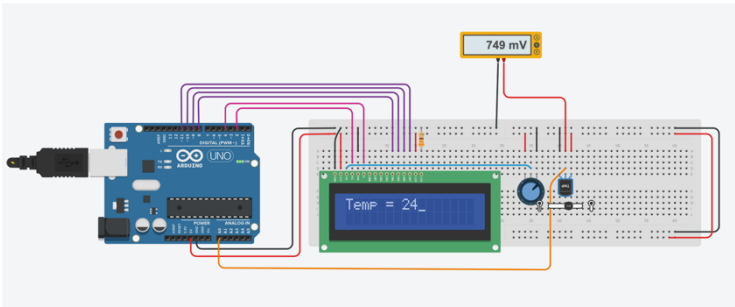


LAB1: LCD Display Interfacing



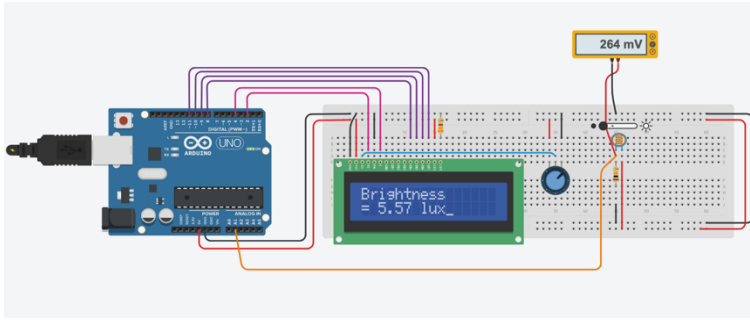
```
1 // C++ code
2 //
3 void commitData(){
4     PORTD |= (1 << PORTD4);
5     _delay_us(10);
6     PORTD &= ~(1 << PORTD4);
7     _delay_us(10);
8 }
9
10 void sendLCDCommand(uint8_t command){
11     //Pull RS Down (Leg D2)
12     PORTD &= ~(1 << PORTD2);
13
14     //Put high nibble(4 bit) of the command
15     PORTB &= 0xF0;
16     PORTB |= command >> 4;
17     commitData();
18
19     //Send low nibble(4 bit) of the command
20     PORTB &= 0xF0;
21     PORTB |= (command & 0x0F);
22
23     commitData();
24 }
25
26 void sendLCDData(uint8_t command){
27     //Pull RS HIGH (Leg D2)
28     PORTD |= (1 << PORTD2);
29
30     //Put high nibble(4 bit) of the command
31     PORTB &= 0xF0;
32     PORTB |= command >> 4;
33     commitData();
34
35     //Send low nibble(4 bit) of the command
36     PORTB &= 0xF0;
37     PORTB |= (command & 0x0F);
38
39     commitData();
40 }
41
42 void lcdDisplayString(char* str){
43     while(*str != '\0')
44     {
45         sendLCDData(*str);
46         str++;
47     }
48 }
49
50 void initLCD(){
51     DDRB |= 0x0F; //0000 1111
52     PORTB &= 0xF0; //clear the last 4 bits to be 0
53     DDRD |= (1 << DDD2) | (1 << DDD4);
54     PORTD &= ~(1 << PORTD2) & ~(1 << PORTD4);
55
56     sendLCDCommand(0x33);
57     sendLCDCommand(0x32);
58     sendLCDCommand(0x28);
59     sendLCDCommand(0x0E);
60
61     // clear
62     sendLCDCommand(0x01);
63     // back to start
64     sendLCDCommand(0x80);}
65
66 void initADC(){
67     // Set reference voltage to AVcc
68     ADMUX |= (1 << REFS0);
69     // Enable ADC with prescaler 128
70     ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
71 }
72
73 void setup()
74 {
75     initLCD();
76     initADC();
77 }
78
79 void loop()
80 {
81     sendLCDCommand(0x01);
82     sendLCDCommand(0x80);
83     //DISPLAY MESSAGE HERE
84     lcdDisplayString("LAB 1");
85     //NEXT LINE
86     sendLCDCommand(0xc0);
87     lcdDisplayString("Hello Kaaa");
88     _delay_ms(1000);
89 }
```

LAB2: AVR ADC Programming



```
1 // C++ code
2 //
3 void commitData(){
4     PORTD |= (1 << PORTD4);
5     _delay_us(10);
6     PORTD &= ~(1 << PORTD4);
7     _delay_us(10);
8 }
9
10 void sendLCDCommand(uint8_t command){
11
12     //Pull RS Down (Leg D2)
13     PORTD &= ~(1 << PORTD2);
14
15     //Put high nibble(4 bit) of the command
16     PORTB &= 0xF0;
17     PORTB |= command >> 4;
18     commitData();
19
20     //Send low nibble(4 bit) of the command
21     PORTB &= 0xF0;
22     PORTB |= (command & 0x0F);
23
24     commitData();
25 }
26
27 void sendLCDData(uint8_t command){
28
29     //Pull RS HIGHHH (Leg D2)
30     PORTD |= (1 << PORTD2);
31
32     //Put high nibble(4 bit) of the command
33     PORTB &= 0xF0;
34     PORTB |= command >> 4;
35     commitData();
36
37     //Send low nibble(4 bit) of the command
38     PORTB &= 0xF0;
39     PORTB |= (command & 0x0F);
40
41     commitData();
42 }
43
44 void lcdDisplayString(char* str){
45     while(*str != '\0')
46     {
47         sendLCDData(*str);
48         str++;
49     }
50 }
51
52 void initLCD(){
53     DDRB |= 0x0F; //0000 1111
54     PORTB &= 0xF0; //clear the last 4 bits to be 0
55     DDRD |= (1 << DDD2) | (1 << DDD4);
56     PORTD &= ~(1 << PORTD2) & ~(1 << PORTD4);
57
58     sendLCDCommand(0x33);
59     sendLCDCommand(0x32);
60     sendLCDCommand(0x28);
61     sendLCDCommand(0x0E);
62     // clear
63     sendLCDCommand(0x01);
64     // back to start
65     sendLCDCommand(0x80);
66 }
67
68
69 void initADC(){
70
71     // Set reference voltage to AVcc
72     ADMUX |= (1 << REFS0);
73     // Enable ADC with prescaler 128
74     ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
75 }
76
77 void setup()
78 {
79     initLCD();
80     initADC();
81     //lcdDisplayString("Hello CPE");
82 }
83
84 char buffer[16];
85
86 void loop()
87 { sendLCDCommand(0x01);
88   sendLCDCommand(0x80);
89
90     // Start conversion
91     ADCSRA |= (1 << ADSC);
92
93     //wait for ADIF
94     while((ADCSRA & (1 << ADSC)));
95
96     // 0-1023 => 0-5V
97     // adcValue to voltage(mV) : (adcValue / 1024.0 * 5)*1000
98     uint16_t adcValue = ADC;
99     uint16_t temp = ((adcValue / 1024.0 * 5000.0) - 500.0) / 10.0;
100     itoa(temp, buffer, 10);
101     lcdDisplayString("Temp = ");
102     lcdDisplayString(buffer);
103
104     _delay_ms(1000);
105 }
```

LAB3: More AVR ADC Programming



```
1 // C++ code
2 //
3 void commitData(){
4     PORTD |= (1 << PORTD4);
5     _delay_us(10);
6     PORTD &= ~(1 << PORTD4);
7     _delay_us(10);
8 }
9
10 void sendLCDCommand(uint8_t command){
11
12     //Pull RS Down (Leg D2)
13     PORTD &= ~(1 << PORTD2);
14
15     //Put high nibble(4 bit) of the command
16     PORTB &= 0xF0;
17     PORTB |= command >> 4;
18     commitData();
19
20     //Send low nibble(4 bit) of the command
21     PORTB &= 0xF0;
22     PORTB |= (command & 0x0F);
23
24     commitData();
25 }
26
27 void sendLCDData(uint8_t command){
28
29     //Pull RS HIGH (Leg D2)
30     PORTD |= (1 << PORTD2);
31
32     //Put high nibble(4 bit) of the command
33     PORTB &= 0xF0;
34     PORTB |= command >> 4;
35     commitData();
36
37     //Send low nibble(4 bit) of the command
38     PORTB &= 0xF0;
39     PORTB |= (command & 0x0F);
40
41     commitData();
42 }
43
44 void lcdDisplayString(char* str){
45     while(*str != '\0')
46     {
47         sendLCDData(*str);
48         str++;
49     }
50 }
51
52 void initLCD(){
53     DDRB |= 0x0F; //0000 1111
54     PORTB &= 0xF0; //clear the last 4 bits to be 0
55     DDRD |= (1 << DDD2) | (1 << DDD4);
56     PORTD &= ~(1 << PORTD2) & ~(1 << PORTD4);
57
58     sendLCDCommand(0x33);
59     sendLCDCommand(0x32);
60     sendLCDCommand(0x28);
61     sendLCDCommand(0x0E);
62     // clear
63     sendLCDCommand(0x01);
64     // back to start
65     sendLCDCommand(0x80);
66 }
67
68 void initADC(){
69     // Set reference voltage to AVcc
70     ADMUX |= (1 << REFS0) | (1 << MUX0);
71     // Enable ADC with prescaler 128
72     ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
73 }
74
75 void setup()
76 {
77     initLCD();
78     initADC();
79 }
80
81 char buffer[16];
82
83 void loop()
84 {
85     sendLCDCommand(0x01);
86     sendLCDCommand(0x80);
87
88     // Start conversion
89     ADCSRA |= (1 << ADSC);
90
91     //wait for ADIF
92     while((ADCSRA & (1 << ADSC)));
93
94     float adcValue = (ADC / 1024.0 * 5);
95     float res = 10000 / ((5.0/adcValue) - 1.0);
96
97     //Not sure about these following equations ka
98     //P Jom pls be kind na ka >____<
99     float lux = res/100.0;
100     //float lux = (1.25 * pow(10,7)) * pow(res,-1.4059);
101
102     dtostrf(lux, 4, 2, buffer);
103     lcdDisplayString("Brightness");
104     sendLCDCommand(0xc0);
105     lcdDisplayString("=");
106     lcdDisplayString(buffer);
107     lcdDisplayString(" lux");
108     _delay_ms(1000);
109 }
```