Microcontrollers
WS2022/23

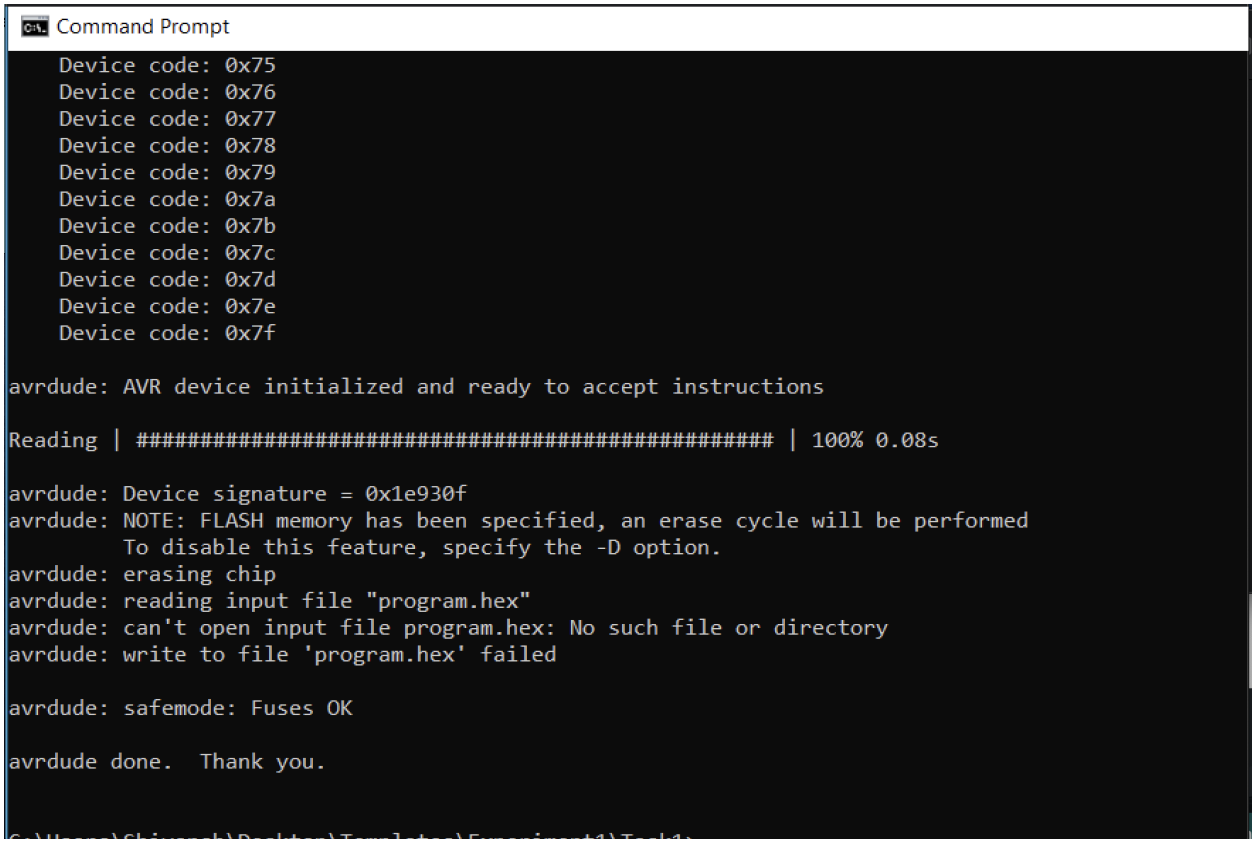
Lab Report 1

Name:	Shivansh Sharma
Matriculation number:	27594
Degree program:	Mechatronic System Engineering
Kit Number:	HSRW MCUOWN
Actual Submission date:	-
Rework submission date:	

Submission Date:

Instructors: Prof. Thomas Grunenberg & Prof. Ivan Volosyak

Task 1:



The 2 green LEDs next to the red LED next to the power input flicker.

Task 2:

Arrangement :

- Pin B4 → Key1
- Pin B5 → Key 2
- Pin B0 → Red LED
- Pin B1 → Yellow LED
- Pin B2 → Green LED
- Pin C0 → poti 1 (potentiometer)
- Pin C3 → Sounder/ Summer

Make flash 1:

Press Key 1 → all LEDs turn on
Press Key 2 → all LEDs turn off

Make flash 2:

LEDs behave in a traffic signal manner where red LED lights up for the longest followed by yellow and green LEDs.

Make flash 3:

2 different sounds are produced by the sounder by pressing the keys and the sound stop with the release of the keys.

Make flash 4:

Press Key 1 → a tune is played

Make flash 5:

Press Key 1 → message displayed on the screen " Hello nice world. I am the atmel Mega88PA microcontroller."

Make flash 6:

Turn Potentiometer (poti 1) → sound produced within the frequency range of 1Hz to 1024Hz.

The two most impressive programs in my opinion were make flash 6 and make flash 2.

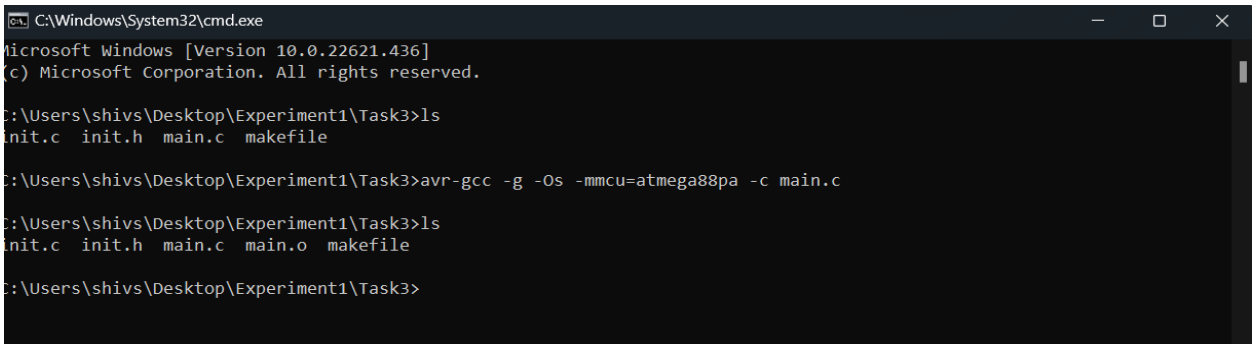
Task 3:

Arrangement :

- Pin D2 → Key1
- Pin D3 → Key 2
- Pin B1 → Red LED

A file “main.o” was created after the execution of the code:

avr-gcc -g -Os -mmcu=atmega88pa -c main.c

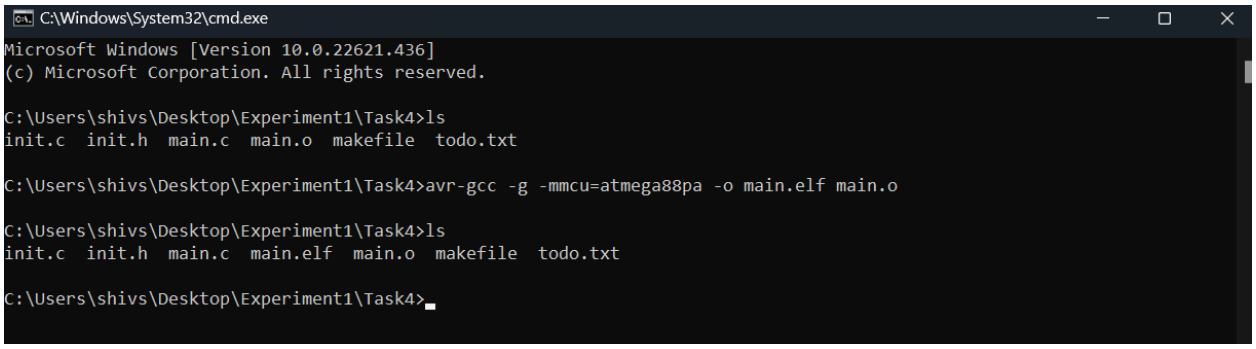


Before the execution

Task 4:

A file “main.elf” was created after the execution of the code:

avr-gcc -g -mmcu=atmega88pa -o main.elf main.o

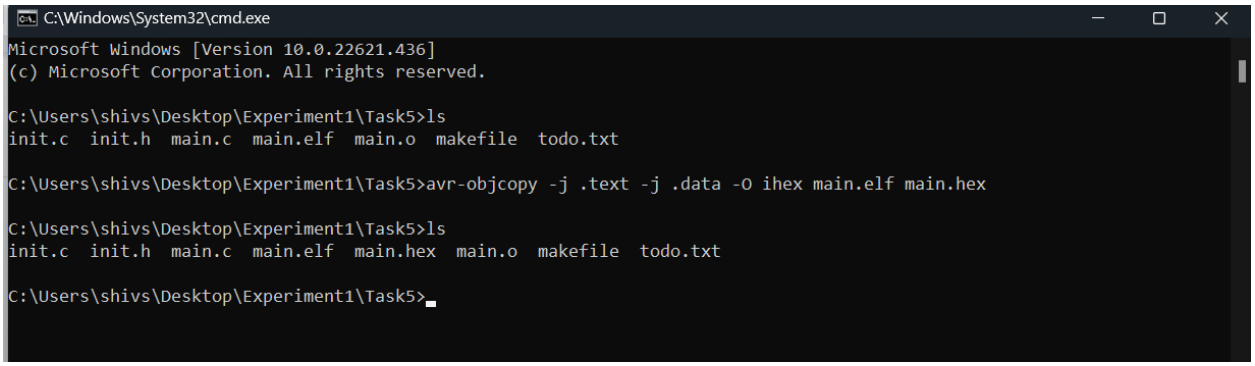


After the execution

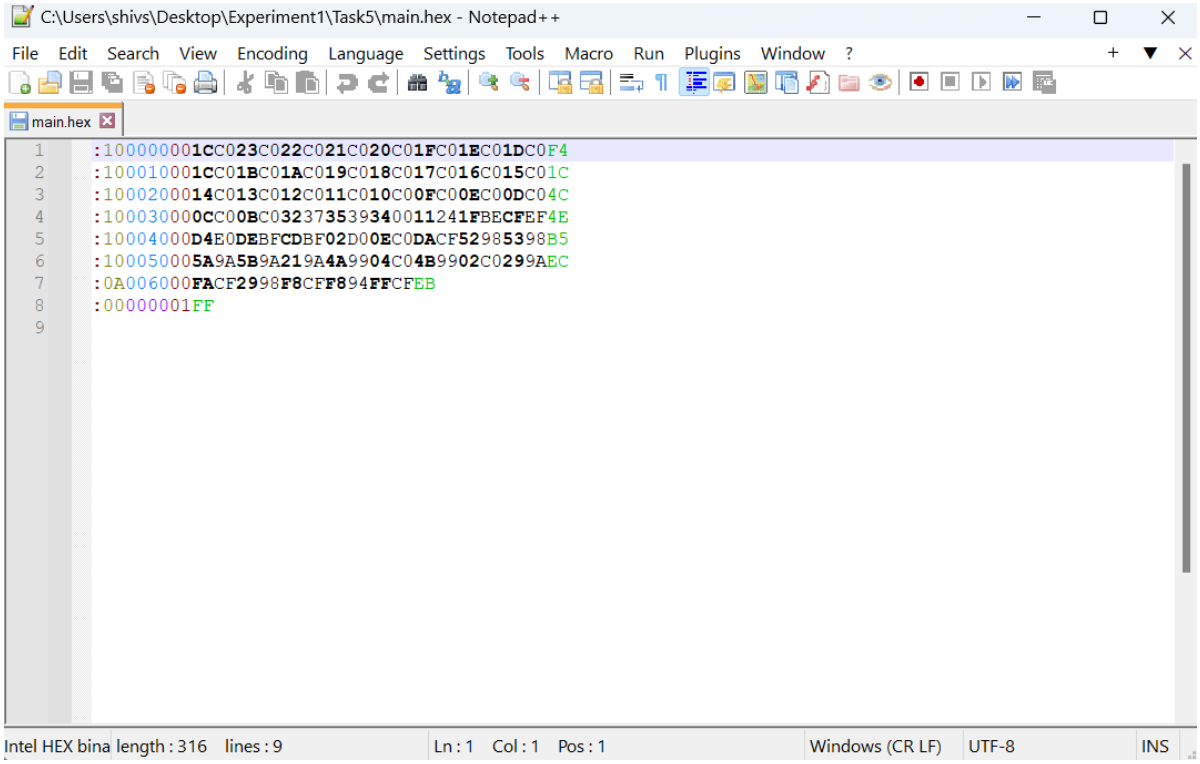
Task 5:

A file “main.hex” was created after the execution of the code:

avr-objcopy -j .text -j .data -O ihex main.elf main.hex



Command Prompt



Hex File

Task 6:

The source code divides into 2 files:

- Main
- Init for i/o initialisation

Task 7:

Arrangement :

- Pin D3 → Key1
- Pin D2 → Key 2
- Pin B1 → Red LED
- Pin B2 → Yellow LED
- Pin B3 → Green LED

NAND:

```
if ((~PIND & (1 << PD2)) && (~PIND & (1 << PD3)))
    PORTB &= ~(1 << PB1);
else
    PORTB |= (1 << PB1);
```

AND:

```
if ((~PIND & (1 << PD2)) && (~PIND & (1 << PD3)))
    PORTB |= (1 << PB2);
else
    PORTB &= ~(1 << PB2);
```

XOR:

```
if (!((~PIND & (1 << PD2))) != (!(~PIND & (1 << PD3))))
    PORTB |= (1 << PB3);
else
    PORTB &= ~(1 << PB3);
```

Task 9:

Arrangement :

- Pin B4 → Key1
- Pin B2 → Red LED

When key 1 is pressed the red led lights up but when the key is let go the red led switches off.

Task 10:

Arrangement :

- Pin B0 → Key1
- Pin B1 → Red LED

A delay function is added within the while loop in this task as compared to the last task.

Task 11:

Arrangement :

- Pin D2 → Key1
- Pin B1 → Red LED

LED is on when the code is executed and with the press and hold of the key the brightness increases but returns back to its initial state once the key is released.

Task 12:

Arrangement :

- Pin C3 → poti 1 (potentiometer)
- Pin B1 → Red LED
- Pin B2 → Yellow LED
- Pin B3 → Green LED

Red, yellow and green LEDs light up ranging from lowest to highest voltage value received by the potentiometer after converting them from analog to digital, respectively.

Task 13:

Arrangement :

- Pin C3 → photo sensor
- Pin B1 → Red LED
- Pin B2 → Yellow LED
- Pin B3 → Green LED

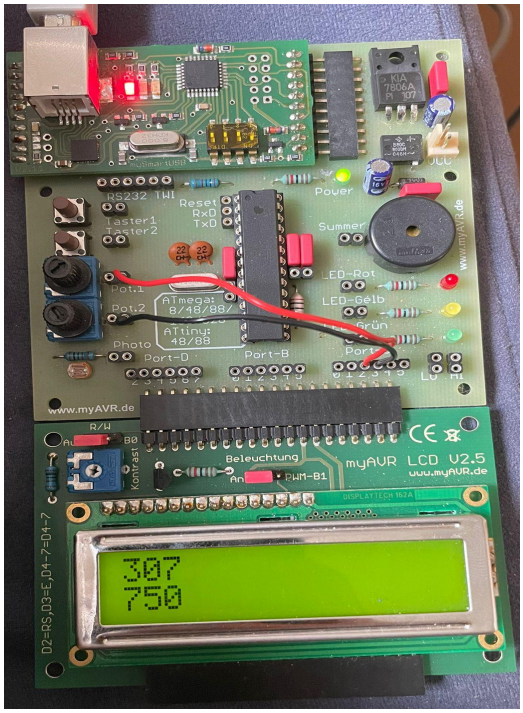
When executed the program converts the analog values received by the photo sensor to digital values using ADC and depending on the intensity each LED is turned on, green turning on at the highest intensity and red at the lowest. A torch was necessary to turn the green LED on. Covering via paper or hand was enough to turn the red LED on.

Task 14 :

Arrangement :

- Pin C3 → poti 1 (potentiometer)
- Pin C2 → poti 2 (potentiometer)

It shows different values of potentiometers.



Output