

Prototype - notat

Tittel: Iterasjon 2

Forfattere: Prosjekt - gruppe 3

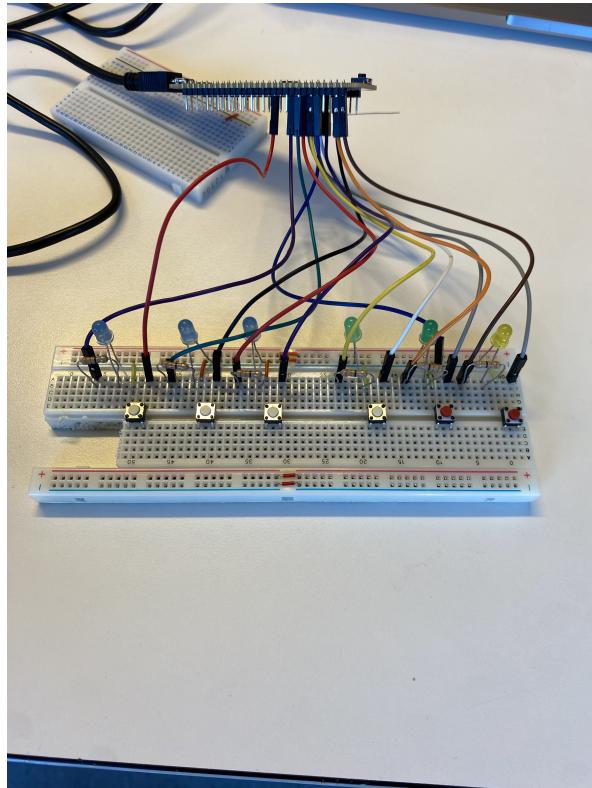
Versjon: 1.0 Dato: 02.10.2024

Innhold

1	Introduksjon	1
2	Metode og testmiljø	2
3	Test resultat	3
4	Begrensninger og tiltak	4
5	Konklusjon	4
6	Vedlegg	4

1 Introduksjon

Denne iterasjonen baserer seg på iterasjon 1. Iterasjon 1 var utelukkende virtuel i labview. Prototype 1 ble modifisert etter input fra statped angående læringsevnen til barn som er født blinde. Prototypen ble forandret fra en lærenehet for lesing til en lærenehet for skriving. Istedentfor å lære å lese alfabetet skal denne prototypen lære å skrive. Tanken er at den skal lese opp en bokstav, når denne bokstaven er riktig skrevet med knapper vil man få en belønning i form at et morsomt lydklipp som representerer bokstaven. Prototypen ble realisert med hardware og testet i liten skala som vist i figur (1).



Figur 1: Oppkobling av prototype 1

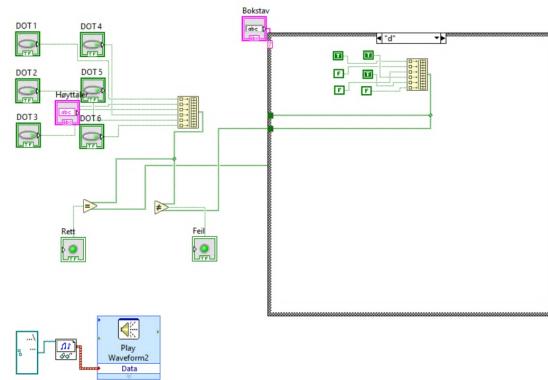
Denne prototypen har som hovedmål å ta for seg implementasjon av knappefunksjonene.

2 Metode og testmiljø

I forbindelse med iterasjon 2 ble det nye designet også testet og implementert i labview før vi begynte med hardware. Labview modell og blokkskjema kan sees i figur xxx



Figur 2: Frontpanel labview

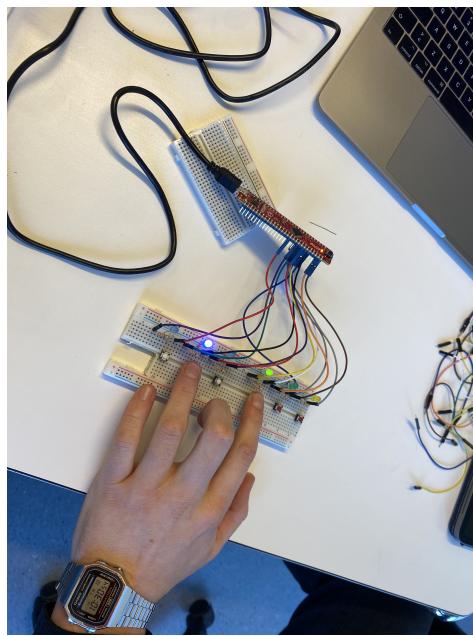


Figur 3: Blokskjema labview

Det ble anvendt en ny mikrokontroller, en AVR128DB48 fra Microchip. Denne ble koblet sammen med 6 knapper på rekke som igjen styrtet hvert sitt led lys. Siden mikrokontrolleren var helt ny for oss samt programmering i C, ble mye tid brukt på å sette seg inn i det nye systemet.

3 Test resultat

Test av prototypen ser ut som i figur (4):



Figur 4: Test av prototype

Figuren viser test av prototypen. Diodene lyser ved at en utgang skrives høy når en knapp blir registrert trykket. Det som er tanken er at det skal simulere signaler man får fra knappene ved skriving.

4 Begrensninger og tiltak

Dette er som nevnt bare en del av det totale systemet som er tenkt. Det mangler en god del som er planlagt implementert ved senere iterasjoner. Nå er knappefunksjonen implementert, videre vil programvare for registrering av riktig bokstav implementert. Samt lagring av lydklipp og avspilling av lydklipp.

5 Konklusjon

Prototypen fungerer som tiltenkt. Knappestrukturen og programvaren fungerer med AVR128DB48. Dette danner grunnlaget for videre iterasjoner med programvare og lyd.

6 Vedlegg

Kode for iterasjon 1:

Listing 1: kode for iterasjon 2

```
1  /*
2   * File:    main.c
3   * Author:  johannes
4   *
5   * Kode for registrering av 6 individuelle knapper
6   *
7   */
8
9 #include "config.h"
10 #include <avr/io.h>
11 #include <avr/interrupt.h>
12 #include <util/delay.h>
13 #include <string.h>
14 #include <stdio.h>
15 #include <stdbool.h>
16 #include "uart.h"
17
18 #define BUTTON_PIN 2 // Knapp tilkoblet PD2, definert som bit 2
19 #define LED_PIN 3    // LED tilkoblet PD3, definert som bit 3
20
21 void setup() {
22     // Initialiserer USART med 9600 baudrate
23     USART3_init(9600);
24
25     // Konfigurerer PD2 som inngang med pull-up for knappen
26     PORTD.DIRCLR = (1 << BUTTON_PIN);      // Sett PD2 som inngang
27     PORTD.PIN2CTRL = PORT_PULLUPEN_bm;      // Aktiver intern pull-up
28     motstand pa PD2
```

```

29     // Konfigurerer PD3 som utgang for LED
30     PORTD.DIRSET = (1 << LED_PIN);           // Sett PD3 som utgang
31     PORTD.OUTCLR = (1 << LED_PIN);           // Start med a sla av LED
32 }
33
34 int main(void) {
35     setup();
36
37     uint8_t button_state = 1;                  // Navarende tilstand til knappen
38     uint8_t last_button_state = 1;             // Forrige tilstand til knappen
39     uint8_t led_state = 0;                     // 0 = av, 1 = pa
40
41     while (1) {
42         // Les knappen (lavt signal betyr at den er trykket)
43         button_state = !(PORTD.IN & (1 << BUTTON_PIN));
44
45         // Sjekker om knappen har gatt fra "ikke trykket" til "trykket"
46         // (fallende flanke)
47         if (button_state && !last_button_state) {
48             USART3_sendString("knapptrykket\n\r"); // Skriv til
49                         Serial Monitor
50
51             // Bytt LED-tilstand
52             if (led_state) {
53                 PORTD.OUTCLR = (1 << LED_PIN); // Sla av LED
54                 led_state = 0;
55             } else {
56                 PORTD.OUTSET = (1 << LED_PIN); // Sla pa LED
57                 led_state = 1;
58             }
59
60             // Oppdaterer siste tilstand til navarende tilstand
61             last_button_state = button_state;
62
63             _delay_ms(50); // Kort forsinkelse for a unnga bouncing
64         }
65     }

```