Podstawy technologi mikroprocesorowej 2

Lab1 - Klawiatura Matrycowa

Imię i Nazwisko:	Remigiusz Mielcarz, Grzegorz Salzburg
Nr indeksu:	252887, 252912

Termin zajęć: dzień tygodnia, godzina:	Środa 14:10-17:10 TP
Numer grupy ćwiczeniowej:	Y03-45f
Data wykonania ćwiczenia:	13.10.2021
Termin do oddania sprawozdania:	27.10.2021
Prowadzący kurs:	Dr inż. Krzysztof Halawa

Spis treści

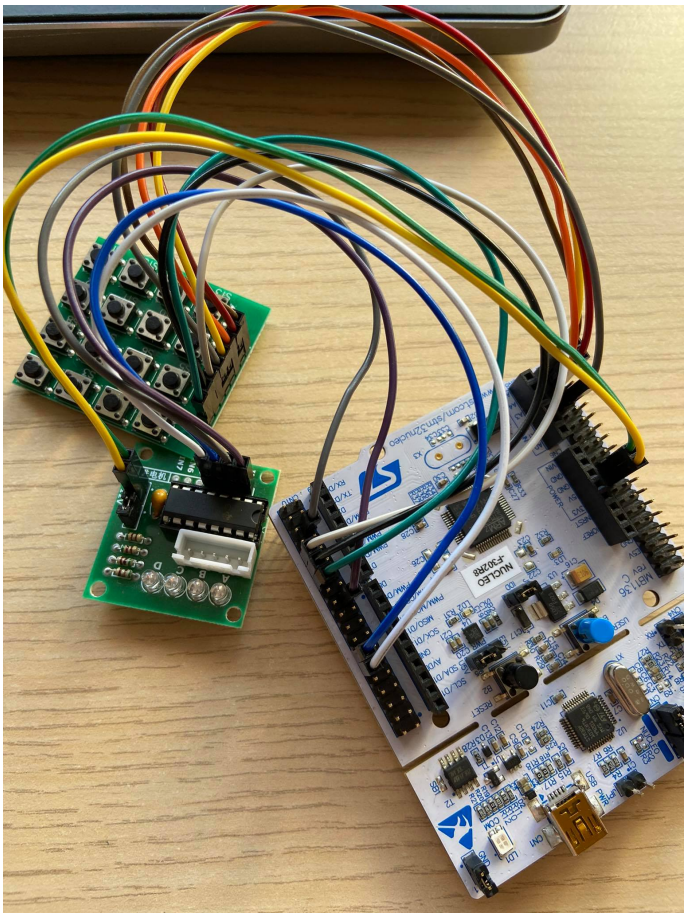
1 Cel projektu

Celem projektu jest zapoznanie się z działaniem mikrokontrolera STM32. Zapoznanie się z obsługą klawiatury matrycowej, podstawowymi funkcjami biblioteki HAL, a także problemem związanym z drganiami styków.

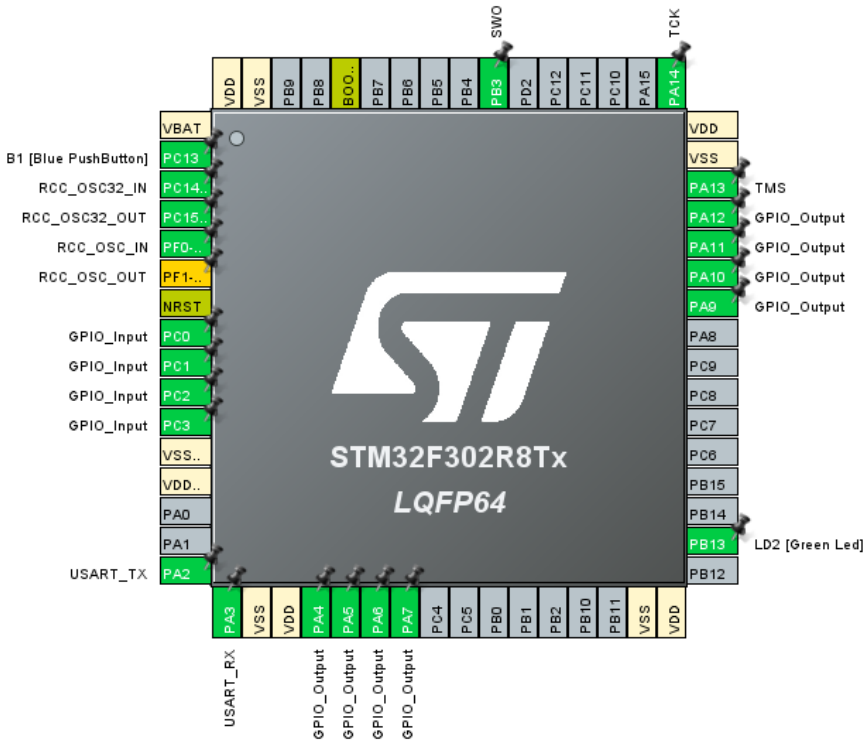
2 Zadania do wykonania

- zapalenie diody LED po naciśnięciu SW1; zgaszenie jej po naciśnięciu SW5
- zmiana stanu LED po naciśnięciu SW3 (należy uwzględnić drgania styków)
- zliczanie naciśnięć SW2 (należy uwzględnić drgania styków). Liczbę naciśnięć należy przedstawić w postaci binarnej za pomocą diód LED wlutowanych w płytkę sterownika silnika krokowego.
- obsługa wszystkich przycisków – numer przycisku należy binarnie pokazać na linijce diodowej

3 Schemat podłączenia



4 Schemat i konfiguracja pinów mikrokontrolera



Rysunek 1: Konfiguracja pinów mikrokontrolera w środowisku CubeIDE.

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO | Single Mapped Signals | RCC | SYS | USART | NVIC

Search Signals

Search (Crtl+F) — Show only Modified Pins

Pin Na...	Signal on ...	GPIO outp...	GPIO mode	GPIO Pull-...	Maximum ...	Fast Mode	User Label	Modified
PB13	n/a	Low	Output Pu...	No pull-up ...	Low	n/a	LD2 [Gree...	☑
PC0	n/a	n/a	Input mode	Pull up	n/a	n/a		☑
PC1	n/a	n/a	Input mode	Pull up	n/a	n/a		☑
PC2	n/a	n/a	Input mode	Pull up	n/a	n/a		☑
PC3	n/a	n/a	Input mode	Pull up	n/a	n/a		☑
PC13	n/a	n/a	External Int...	No pull-up ...	n/a	n/a	B1 [Blue P...	☑

Rysunek 2: Konfiguracja pinów - szczegóły

- Piny wejściowe (GPIO_Input) z włączonymi rezystorami podciągającymi (ang. Pull up)

  - PC0, PC1, PC2, PC3,

- Piny wyjściowe (GPIO_Output)

  - PA4, PA5, PA6, PA7, PA9, PA10, PA11, PA12,

5 Kod programu

```c
#include "main.h"

UART_HandleTypeDef huart2;

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART2_UART_Init(void);

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

void kolumna_1(void)
{
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);   // Ustaw kolumne 1 na 0
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);     // Ustaw kolumne 2 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);     // Ustaw kolumne 3 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);     // Ustaw kolumne 4 na 1
}

void kolumna_2(void)
{
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);   // Ustaw kolumne 2 na 0
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);     // Ustaw kolumne 1 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);     // Ustaw kolumne 3 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);     // Ustaw kolumne 4 na 1
}

void kolumna_3(void)
{
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);   // Ustaw kolumne 3 na 0
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);     // Ustaw kolumne 1 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);     // Ustaw kolumne 2 na 1
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);     // Ustaw kolumne 4 na 1
}

uint8_t klawisz_wcisniety(void)
{
        /* Jesli zosta  wcisniety przycisk*/
        if (HAL_GPIO_ReadPin(wcisniety_GPIO_PORT, wcisniety_GPIO_Pin) == 0)
        {
                HAL_Delay(80);          // 80 ms opoznienia
                /* jesli klawisz nadal wcisniety */
                if (HAL_GPIO_ReadPin(wcisniety_GPIO_PORT, wcisniety_GPIO_Pin) == 0)
                        return 1;       // Zwroc 1
```

```
46          }
47
48          return 0;       // Zwroc 0 je li klawisz tylko zosta  wcisniety
49  }
50
51  // zwraca pozycje wcisnietego przycisku w kolumnie (domyslnie "0")
52  // odczytuje stany wierszy
53  int ktory_przycisk(void)
54  {
55          return !HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_0) * 1 +
56                          !HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_1) * 2 +
57                          !HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_2) * 3 +
58                          !HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_3) * 4;
59  }
60
61  void zapal_diody(liczba_wcisniec)
62  {
63            if (liczba_wcisniec == 1)
64            {
65                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
66                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
67                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
68                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
69            }
70            else if (liczba_wcisniec == 2)
71            {
72                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
73                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
74                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
75                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
76            }
77            else if (liczba_wcisniec == 3)
78            {
79                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
80                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
81                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
82                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
83            }
84            else if (liczba_wcisniec == 4)
85            {
86                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
87                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
88                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
89                    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
90            }
91            else if (liczba_wcisniec == 5)
92            {
```

```
93                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
94                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
95                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
96                  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
97          }
98          else if (liczba_wcisniec == 6)
99          {
100                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
101                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
102                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
103                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
104         }
105         else if (liczba_wcisniec == 7)
106         {
107                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
108                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
109                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
110                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
111         }
112         else if (liczba_wcisniec == 8)
113         {
114                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
115                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
116                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
117                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
118         }
119         else if (liczba_wcisniec == 9)
120         {
121                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
122                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
123                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
124                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
125         }
126         else if (liczba_wcisniec == 10)
127         {
128                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
129                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
130                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
131                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
132         }
133         else if (liczba_wcisniec == 11)
134         {
135                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
136                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
137                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
138                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
139         }
```

```
140             else if (liczba_wcisniec == 12)
141             {
142                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
143                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
144                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
145                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
146             }
147             else if (liczba_wcisniec == 13)
148             {
149                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
150                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
151                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
152                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
153             }
154             else if (liczba_wcisniec == 14)
155             {
156                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
157                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
158                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
159                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
160             }
161             else if (liczba_wcisniec == 15)
162             {
163                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
164                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
165                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
166                     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);
167             }
168 }
169 /* USER CODE END 0 */
170
171 /**
172   * @brief  The application entry point.
173   * @retval int
174   */
175 int main(void)
176 {
177   /* MCU Configuration--------------------------------------------------------*/
178   /* Reset of all peripherals, Initializes the Flash interface and the Systick.*/
179   HAL_Init();
180
181   /* Configure the system clock */
182   SystemClock_Config();
183
184   /* Initialize all configured peripherals */
185   MX_GPIO_Init();
186   MX_USART2_UART_Init();
```

```
187
188   /* USER CODE BEGIN 2 */
189
190 //  POCZ ZAD 1 ***************************************************************
191         kolumna_1();
192 //  KONIEC ZAD 1 *************************************************************
193
194 //  POCZ ZAD 2 ***************************************************************
195         kolumna_3();
196 //  KONIEC ZAD 2 *************************************************************
197
198 //  POCZ ZAD 3 ***************************************************************
199         kolumna_2();
200         int liczba_wcisniec = 0;
201 //  KONIEC ZAD 3 *************************************************************
202
203 //  POCZ ZAD 4 ***************************************************************
204 //stany_kolumn - tabela sekwencji stanow kolumn klawiatury
205   const uint16_t stany_kolumn[] = {
206                   (GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7),
207                   (GPIO_PIN_4 | GPIO_PIN_6 | GPIO_PIN_7),
208                   (GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_7),
209                   (GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6)};
210
211   int i = 0;
212   int wartosc_przycisku = 0;
213
214 //  KONIEC ZAD 4 *************************************************************
215
216   /* USER CODE END 2 */
217
218   /* Infinite loop */
219   /* USER CODE BEGIN WHILE */
220
221   while (1)
222   {
223
224 //  POCZ ZAD 1 ***************************************************************
225 //  1) zapalenie diody LED po nacisnieciu SW1; zgaszenie jej po nacisnieciu SW5
226
227         if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_0) == 0)           // PC0 - wiersz 1
228         {
229                   HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);
230           }
231           else if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_1) == 0)    // PB0 - wiersz 2
232           {
233                   HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);
```

```
234           }
235 //  KONIEC ZAD 1 ********************************************************************
236
237 //  POCZ ZAD 2 ********************************************************************
238 //  2) zmiana stanu LED po naci ni ciu SW3 (nale y uwzglednic drgania stykow)
239
240         if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_0) == 0)
241         {
242                 HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
243                 HAL_Delay(40);
244                 while(klawisz_wcisniety())
245                 {
246                 }
247           }
248 //  KONIEC ZAD 2 ********************************************************************
249
250 //  POCZ ZAD 3 ********************************************************************
251 //  3) zliczanie naci ni    SW2 (nalezy uwzglednic drgania stykow).
252 //  wynik nale y przedstawic binarnie na linijce diodowej
253
254         if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_0) == 0)
255         {
256                 liczba_wcisniec++;
257                 HAL_Delay(40);
258                 while(klawisz_wcisniety())
259                 {
260                 }
261         }
262         zapal_diody(liczba_wcisniec);
263
264 //  KONIEC ZAD 3 ********************************************************************
265
266 //  POCZ ZAD 4 ********************************************************************
267 //  4)obsluga wszystkich przyciskow     numer przycisku binarnie na linijce diodowej
268
269         /*wyzerowanie kolumn*/
270         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7, 0);
271         /*ustawienie sekwencji stanow (z tabeli stanow)*/
272         HAL_GPIO_WritePin(GPIOA, stany_kolumn[i++], 1);
273         if(i > 4) i = 0;
274         int wiersz = ktory_przycisk(); //numer wcisnietego przycisku w sprawdzanej kolumnie
275
276         if(wiersz > 0) {
277         HAL_Delay(40); //odczekanie, az wygasna drgania stykow
278         wartosc_przycisku = (wiersz - 1) * 4 + i; //obliczenie wartosci wcisnietego przycisku
279         zapal_diody(wartosc_przycisku);
280
```

```
281         while ( ktory_przycisk() > 0); //oczekiwanie na puszczenie przycisku
282         HAL_Delay(40); //odczekanie, az wygasna drgania stykow
283         }
284         else wartosc_przycisku = 0;
285
286 //  KONIEC ZAD 4 ************************************************************************
287
288     /* USER CODE END WHILE */
289     /* USER CODE BEGIN 3 */
290   }
291   /* USER CODE END 3 */
292 }
```

- Powyższy program zawiera wszystkie 4 zadania. Do poprawnego działania każdego z zadań, należy skomentować pozostałe 3 programy (każde oznaczone komentarzami).
- Do zadania 1 korzystamy z przycisków z 1 kolumny (SW1 i SW5), dlatego przy użyciu funkcji void kolumna_1 dzięki której ustawiamy stan 0 na potrzebną nam kolumnę. Reszta zadania polega na zwykłych instrukcjach if, dzięki którym wyłączamy i włączamy diodę LED (LD2)
- Do zadania 2 korzystamy z analogicznej funkcji void kolumna_3, która ustawia nam stan 0 na 3 kolumnie dla naszego switcha SW3. Zadanie zawiera zmianę stanu LED, przy uwzględnieniu drgania styków. Niwelacja drgań styków jest realizowana poprzez oczekiwanie stanu klawisza po zmianie. W momencie przytrzymywania przycisku, program czeka na jego puszczenie.
- Do zadania 3 korzystamy analogicznie z funkcji void kolumna_2, dla przycisku SW2. Program zlicza naciśnięcia przycisku SW2, przedstawiając tą wartość na linijce diodowej, dzięki funkcji void zapal_diody(liczba_wcisniec), która dla liczb od 0 do 15, zapala odpowiednie sekwencje na linijce diodowej.
- Zadanie 4 polega na ustawianiu stanu niskiego na kolejnych kolumnach i przeszukiwaniu wierszy w poszukiwaniu stanu niskiego, który pozwoli nam uzyskać współrzędne przycisku, który został właśnie wciśnięty. Współrzędne te przeliczane są na wartość od 1 do 15, które odpowiadają odpowiednim switch'om. Wyświetlanie tej wartości na linijce diodowej, polega na użyciu funkcji, użytej w poprzednim zadaniu (void zapal_diody(liczba_wcisniec))

Wykonano w systemie LaTeX