



AGH

Sprawozdanie z wykonania projektu systemu podkładki
alarmowej

Sonia Orlikowska

6 czerwca 2021

Wydział Inżynierii Metali i Informatyki Przemysłowej
Kierunek: Informatyka Techniczna
nr albumu: 287946

Spis treści

1	Streszczenie	1
2	Model teoretyczny - Czujnik ultradźwiękowy	1
3	Opis i budowa układu	2
4	Oprogramowanie	3
5	Realizacja projektu	7

1 Streszczenie

W ramach przedmiotu zrealizowałem prosty system alarmowy w oparciu o ultradźwiękowy pomiar odległości. W tym projekcie system został zastosowany jako podkładka z alarmem. Użytkownik może odłożyć dany sprzęt na taką podkładkę uzbroić alarm i zostawić sprzęt. W chwili podniesienia przedmiotu z podkładki załącza się alarm (sygnał dźwiękowy buzzera) który należy wyłączyć podając czterocyfrowy PIN. Alarm dźwiękowy będzie aktywny aż do czasu podania poprawnego kodu PIN. Dodatkowo za pomocą wyświetlacza i pinpadu można zmienić kod pin na inny. System jest o tyle ciekawy i rozwojowy, że przy bardzo niewielkiej ingerencji w kod można zmienić jego zastosowanie nie zmieniając konfiguracji układu. Taki układ można z łatwością przekształcić na prosty czujnik ruchu z alarmem lub system alarmowy informujący o odległości np. system wspomagający parkowanie. Jedną z największych wad układu jest zbyt duża obudowa, która była wymuszoną niestety brakiem odpowiednich narzędzi. Ciekawym kierunkiem rozbudowy układu i jego faktycznego zastosowania było by zaprojektowanie obudowy z tworzywa, zapewnienie komunikacji bezprzewodowej i zasilania na baterię, aby urządzenie mogło być bardziej przenośne. Ciekawym rozwiązaniem byłoby również stworzenie systemu powiadomień np. SMS informującego o załączeniu alarmu.

2 Model teoretyczny - Czujnik ultradźwiękowy

Podstawowym komponentem bez, którego cały układ nie miałby prawa funkcjonować jest ultradźwiękowy czujnik ruchu. Czujnik zastosowany w projekcie to HC-SR04. Jego zakres pomiarowy to 2-200 cm. Moduł ultradźwiękowy HC-SR04 ma 4 piny, GND VCC, Trig i Echo. Piny GND i VCC modułu muszą być podłączone odpowiednio do uziemienia i pinów 5V , piny wyzwalający i echo do dowolnych pinów cyfrowego wejścia / wyjścia. Wykres czasowy generacji i odbioru impulsów przedstawiono na Rysunku 1. Aby wygenerować ultradźwięki, należy ustawić Trig w stan wysoki na conajmniej $10 \mu\text{s}$. To wygeneruje 8-cyklowy impuls dźwiękowy o częstotliwości 40kHz. Ten odbity od przeszkody zostanie odebrany w pinie Echo. Gdy czujnik wykryje ultradźwięki z odbiornika, ustawi pin Echo na wysoki (5V) i opóźni o okres (szerokość), który jest proporcjonalny do odległości. Na wyjściu pinu Echo otrzymamy czas w mikrosekundach, w którym fala dźwiękowa wróciła do sonaru.

Ten czas możemy zapisać jako:

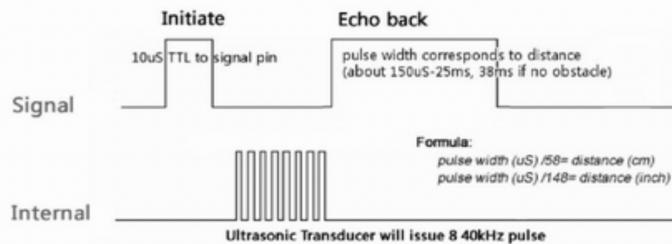
$$\text{czas} = \text{szerokość pulsu Echo w } [\mu\text{s}]$$

W celu obliczenia odległości obiektu od czujnika należy zastosować wzór (1) na prędkość dźwięku:

$$s = \frac{t}{2} \cdot 0.034 \quad (1)$$

$$[cm] = [\mu s] \left[\cdot \frac{cm}{\mu s} \right]$$

Rysunek 1: Schemat generowania i odbierania sygnału przez czujnik [?]



Parametry czujnika HC-SR04:

- Zasilanie: +5V DC
- Prąd spoczynkowy: < 2mA
- Prąd roboczy: 15mA
- Efektywny kąt pomiaru: < 15°
- Zakres pomiaru: 2- 200cm
- Rozdzielcość pomiaru ±0.3cm
- Całkowity kąt pomiaru 30°
- Długość sygnału wyzwalającego: 10 μ s
- Wymiary: 45mm x 20mm x 15mm

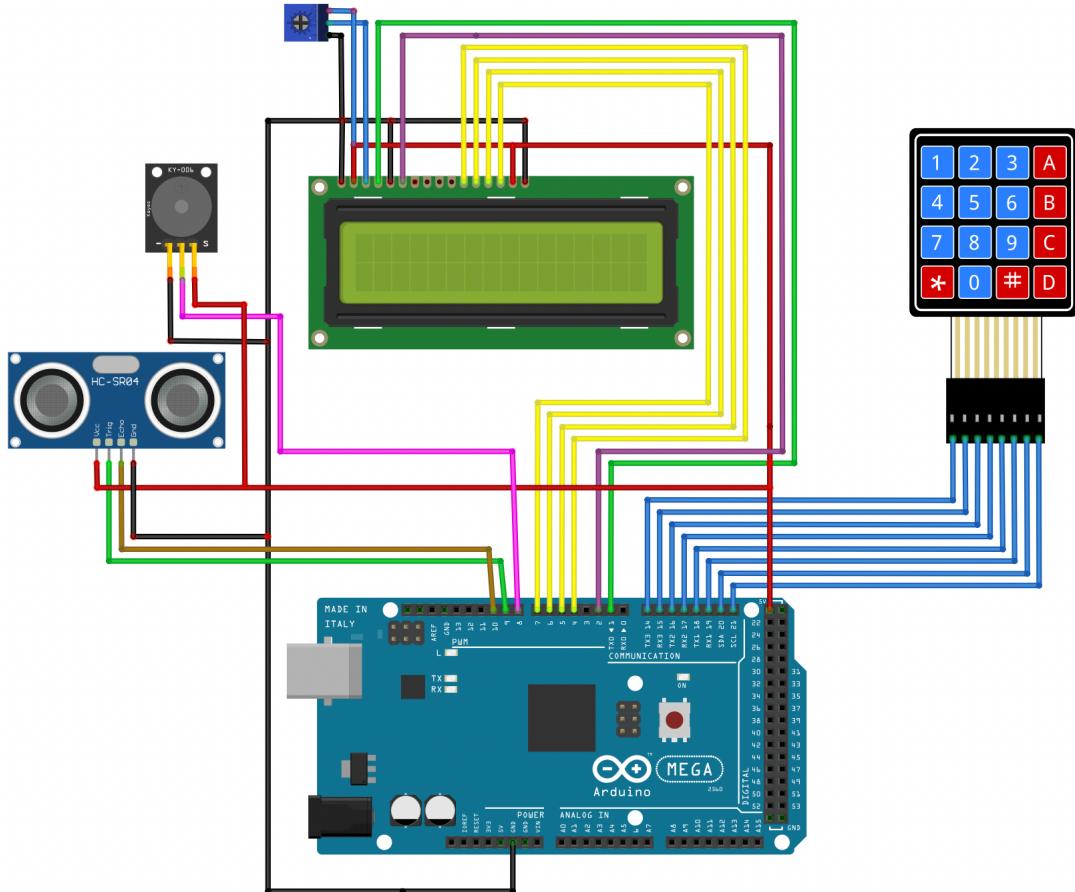
3 Opis i budowa układu

Schemat układu znajduje się na Rysunku 1, do narysowania układu wykorzystano program . Do zbudowania układu wykorzystano następujące elementy. Układ nie był lutowany, w celu połączeń posłużyłam się płytą stykową.

- Płytki Arduino Mega 2560 Rev3 - A000067
- Ultradźwiękowy czujnik odległości HC-SR04 2-200cm
- Wyświetlacz LCD 2x16 znaków
- Buzzer
- Klawiatura numeryczna membranowa samoprzylepna - 16 klawiszy
- Potencjometr montażowy leżący

- Przewody połączeniowe
- Płytki stykowa

Rysunek 2: Schemat układu połączeń



4 Oprogramowanie

Na następnych stronach umieściłem kompletny kod którego użyłam podczas wykonywania projektu. Kod jest opatrzony komentarzami w miejscach w, których wydawało się to być przydatne. W implementacji musiłem skorzystać z dwóch bibliotek `include <LiquidCrystal.h>` oraz `include <Keypad.h>` aby móc korzystać z wyświetlacza i pinpadu. Program składa się z funkcji `void setup()`, `void loop()`, `void enterPassword()`, `long getDistance()`. Funkcja `enterPassword()` ma za zadanie sprawdzać poprawność wpisanego hasła za pomocą pinpadu, funkcja `getDistance()` ma za zadanie pobrać dane z czujnika odległości.

alarm §

```
1 // potrzebujemy dwie biblioteki do obsługi wyświetlacza LCD oraz keypadu
2 #include <LiquidCrystal.h>
3 #include <Keypad.h>
4
5 // zdefiniowanie pinów przyłączeniowych dla buzzera i czujnika ultradźwiękowego
6 #define buzzer 8
7 #define trigPin 9
8 #define echoPin 10
9
10 //zdefiniowanie wykorzystywanych zmiennych
11 long duration;
12 int distance, initialDistance, currentDistance, i;
13 boolean showMenu = true;
14 String password = "1234";
15 String tempPassword;
16 boolean activated = false;
17 boolean isActivated;
18 boolean activateAlarm = false;
19 boolean alarmActivated = false;
20 boolean enteredPassword;
21 boolean passChangeMode = false;
22 boolean passChanged = false;
23
24 //ustawienia keypadu
25 const byte ROWS = 4; //4 rzędy klawiatury
26 const byte COLS = 4; //4 kolumny klawiatury
27 char keypressed;
28
29 //zdefiniowanie macierzy symboli klawiatury
30 char keyMap[ROWS][COLS] = {
31     {'1', '2', '3', 'A'},
32     {'4', '5', '6', 'B'},
33     {'7', '8', '9', 'C'},
34     {'*', '0', '#', 'D'}
35 };
36
37 byte rowPins[ROWS] = {14, 15, 16, 17}; //Przypisanie numerów pinów do każdego rzędu pinpadu
38 byte colPins[COLS] = {18, 19, 20, 21}; //Przypisanie numerów pinów do każdej kolumny pinpadu
39
40 Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, ROWS, COLS); //obiekt Keypad
41
42 LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // LC obiekt. Parametry: (rs, enable, d4, d5, d6, d7)
43
44 void setup() {
45     lcd.begin(16, 2);
46     pinMode(buzzer, OUTPUT); // Ustawia buzzer jako output
47     pinMode(trigPin, OUTPUT); // Ustawia trigPin jako output
48     pinMode(echoPin, INPUT); // Ustawia echoPin jako input
49 }
50
51 void loop() {
52     if (activateAlarm) {
53         lcd.clear();
54         lcd.setCursor(0, 0);
55         lcd.print("Alarm bedzie");
56         lcd.setCursor(0, 1);
57         lcd.print("aktywowany za ");
58
59         int countdown = 5; // 5 sekund do aktywacji alarmu
60         while (countdown != 0) {
61             lcd.setCursor(15, 1);
62             lcd.print(countdown);
63             countdown--;
64             tone(buzzer, 1300, 100);
65             delay(1000);
66         }
67         lcd.clear();
68         lcd.setCursor(0, 0);
69         initialDistance = getDistance();
70         lcd.print("Alarm Aktywny!!!");
71         activateAlarm = false;
72         alarmActivated = true;
73     }
74
75     //Aktywowanie alarmu dźwiękowego gdy przedmiot zostanie zabrany
76     if (alarmActivated == true) {
77         currentDistance = getDistance();
78         if (currentDistance > 2 && currentDistance < 200) {
79             tone(buzzer, 1000);
80             enterPassword();
81         }
82     }
83 }
```

```

83 //Pierwsza akcja na wyświetlaczu
84 if (!alarmActivated) {
85     if (showMenu) {
86         lcd.clear();
87         lcd.setCursor(0, 0);
88         lcd.print("A Aktywuj Alarm");
89         lcd.setCursor(0, 1);
90         lcd.print("B Zmien Haslo");
91         showMenu = false;
92     }
93     keypressed = myKeypad.getKey();
94     if (keypressed == 'A') { //Po kliknięciu 'A' Alarm jest aktywowany
95         //tone(buzzer, 1000, 200);
96         activateAlarm = true;
97     }
98     else if (keypressed == 'B') {
99         lcd.clear();
100        int i = 1;
101        tone(buzzer, 2000, 100);
102        tempPassword = "";
103        lcd.setCursor(0, 0);
104        lcd.print("Obecne Haslo");
105        lcd.setCursor(0, 1);
106        lcd.print(">");
107        passChangeMode = true;
108        passChanged = true;
109        while (passChanged) {
110            keypressed = myKeypad.getKey();
111            if (keypressed != NO_KEY) {
112                if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
113                    keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
114                    keypressed == '8' || keypressed == '9') {
115                    tempPassword += keypressed;
116                    lcd.setCursor(i, 1);
117                    lcd.print("*");
118                    i++;
119                    tone(buzzer, 2000, 100);
120                }
121            }
122        }
123        if (i > 5 || keypressed == '#') {
124            tempPassword = "";
125            i = 1;
126            lcd.clear();
127            lcd.setCursor(0, 0);
128            lcd.print("Obecne Haslo");
129            lcd.setCursor(0, 1);
130            lcd.print(">");
131        }
132        if (keypressed == '*') {
133            i = 1;
134            tone(buzzer, 2000, 100);
135            if (password == tempPassword) {
136                tempPassword = "";
137                lcd.clear();
138                lcd.setCursor(0, 0);
139                lcd.print("Ustaw nowe haslo");
140                lcd.setCursor(0, 1);
141                lcd.print(">");
142                while (passChangeMode) {
143                    keypressed = myKeypad.getKey();
144                    if (keypressed != NO_KEY) {
145                        if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
146                            keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
147                            keypressed == '8' || keypressed == '9') {
148                            tempPassword += keypressed;
149                            lcd.setCursor(i, 1);
150                            lcd.print("*");
151                            i++;
152                            tone(buzzer, 2000, 100);
153                        }
154                    }
155                    if (i > 5 || keypressed == '#') {
156                        tempPassword = "";
157                        i = 1;
158                        tone(buzzer, 2000, 100);
159                        lcd.clear();
160                        lcd.setCursor(0, 0);
161                        lcd.print("Obecne Haslo:");
162                        lcd.setCursor(0, 1);
163                        lcd.print(">");
164                    }

```

```

164 }
165     if (keypressed == '*') {
166         i = 1;
167         tone(buzzer, 2000, 100);
168         password = tempPassword;
169         passChangeMode = false;
170         passChanged = false;
171         showMenu = true;
172     }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180
181 void enterPassword() {
182     int k = 5;
183     tempPassword = "";
184     activated = true;
185     lcd.clear();
186     lcd.setCursor(0, 0);
187     lcd.print(" *** ALARM *** ");
188     lcd.setCursor(0, 1);
189     lcd.print("Pass>");
190     while (activated) {
191         keypressed = myKeypad.getKey();
192         if (keypressed != NO_KEY) {
193             if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
194                 keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
195                 keypressed == '8' || keypressed == '9') {
196                 tempPassword += keypressed;
197                 lcd.setCursor(k, 1);
198                 lcd.print("*");
199                 k++;
200             }
201         }
202     }
203     if (k > 9 || keypressed == '#') {
204         tempPassword = "";
205         k = 5;
206         lcd.clear();
207         lcd.setCursor(0, 0);
208         lcd.print(" *** ALARM *** ");
209         lcd.setCursor(0, 1);
210         lcd.print("Pass>");
211     }
212     if (keypressed == '*') {
213         if (tempPassword == password) {
214             activated = false;
215             alarmActivated = false;
216             noTone(buzzer); //wyłączenie sygnału alarmu po wprowadzeniu poprawnego hasła
217             showMenu = true;
218         } else if (tempPassword != password) {
219             lcd.setCursor(0, 1);
220             lcd.print("Zle hasło!!!");
221             delay(2000);
222             lcd.clear();
223             lcd.setCursor(0, 0);
224             lcd.print(" !!! ALARM !!! ");
225             lcd.setCursor(0, 1);
226             lcd.print("Pass>");
227         }
228     }
229 }
230 }
231 // Funkcja do odczytu odległości z czujnika ultradźwięków
232 long getDistance() {
233
234     //zerowanie trigPin
235     digitalWrite(trigPin, LOW);
236     delayMicroseconds(2);
237
238     //trigPin wprowadzony w stan HIGH na 10 micro sekund
239     digitalWrite(trigPin, HIGH);
240     delayMicroseconds(10);
241     digitalWrite(trigPin, LOW);
242
243     digitalWrite(trigPin, LOW);
244
245     //Odczyt z input
246     duration = pulseIn(echoPin, HIGH);
247
248     //odległość podana w cm
249     distance = duration / 58;
250
251 }

```

5 Realizacja projektu

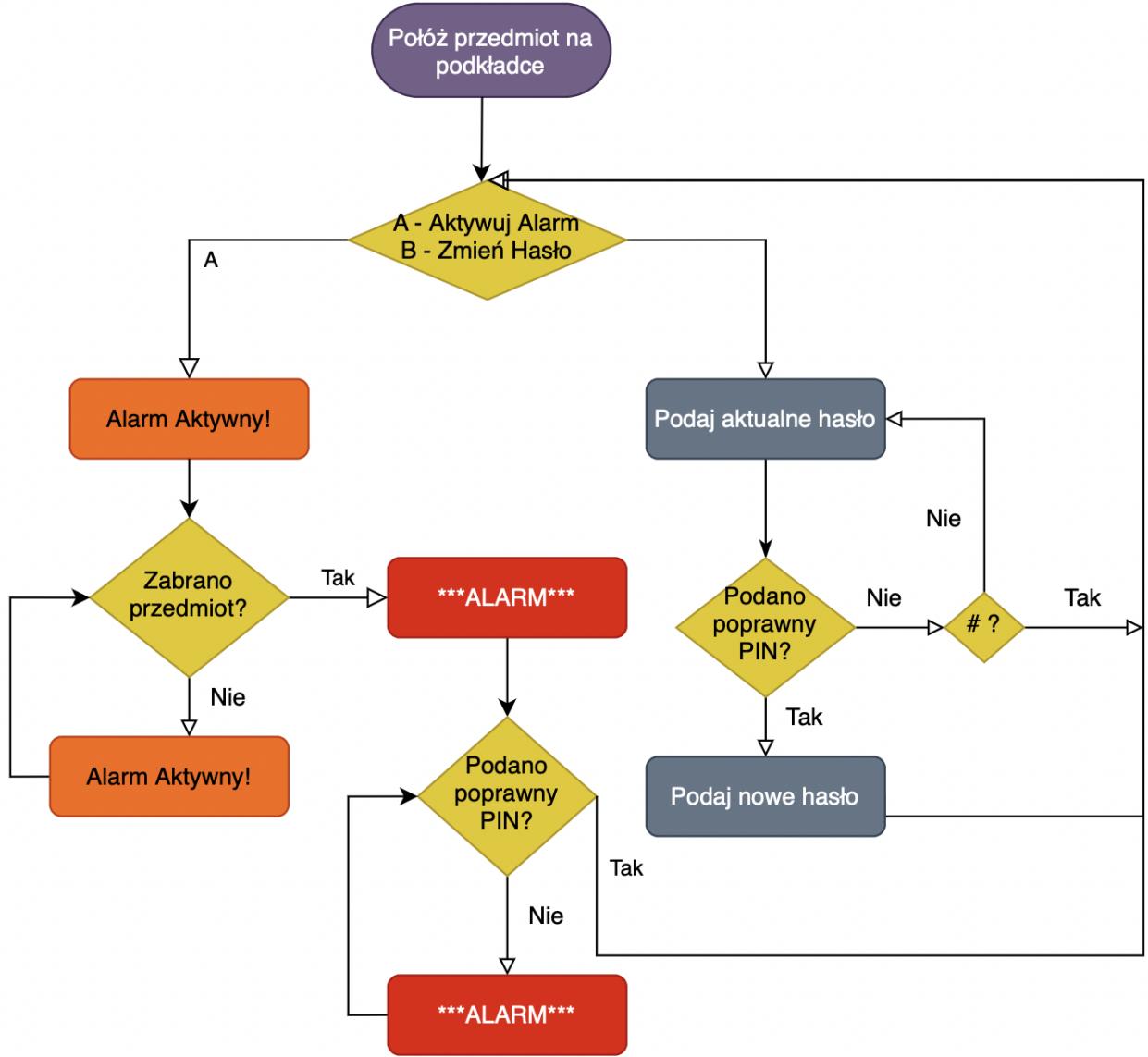
Projekt udało się zrealizować, efekt finalny przedstawia Rysunek 3. Wszystkie komponenty włożono do kartonowego pudełka, czujnik oraz wyświetlacz LCD zostały przytwierdzone za pomocą taśmy przylepnej. Układ nie był lutowany, połączenia zostały wykonane bezpośrednio lub pośrednio korzystając z płytki stykowej.

Rysunek 3: Efekt końcowy



Na Rysunku 4 przedstawiono uproszczony schemat działania układu, zaprezentowano również jakie funkcje zostały zaimplementowane.

Rysunek 4: Uproszczony schemat działania układu



- Układ pozwala na uzbrojenie alarmu poprzez wciśnięcie klawisza A po wcześniejszym ułożeniu przedmiotu na podkładce
- Po zabraniu przedmiotu z podkładki natychmiast uruchamia się sygnał alarmowy, który można wyłączyć tylko za pomocą aktualnego hasła
- Układ pozwala na zmianę 4 cyfrowego hasła
- Układ sprawdza czy hasło wprowadzone poprzez pinpad jest poprawne
- Potwierdzenie czynności odbywa się po wciśnięciu symbolu ‘*’
- Po wprowadzeniu więcej niż 4 znaków lub symbolu ‘#’ następuje wyczyszczenie ekranu i można ponownie wpisać hasło