
PING PONG

Grupa numer 16
Informatyka
II rok
Rok akademicki 2017/2018

Autorzy:

KOLANO GRZEGORZ
WIATER MATEUSZ
WOLSKI ŁUKASZ

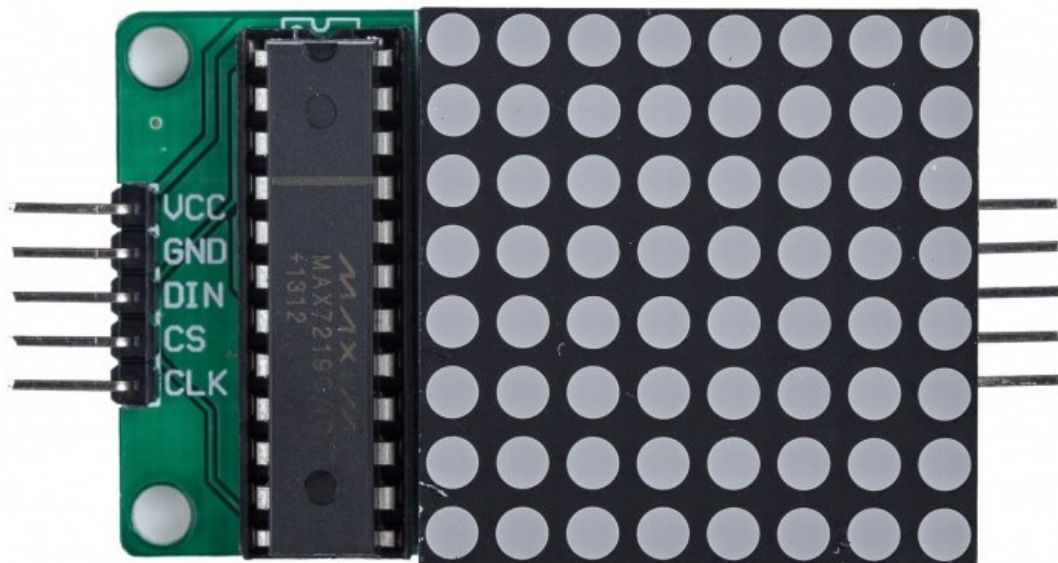
11 września 2018

1 Użyte moduły

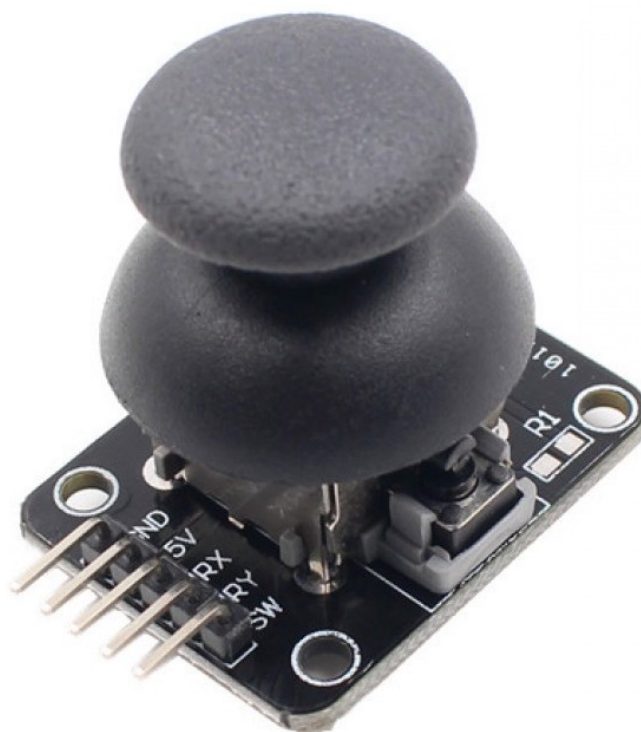
1.1 Arduino Uno



1.2 Matryca LED 8x8 ze sterownikiem MAX7219

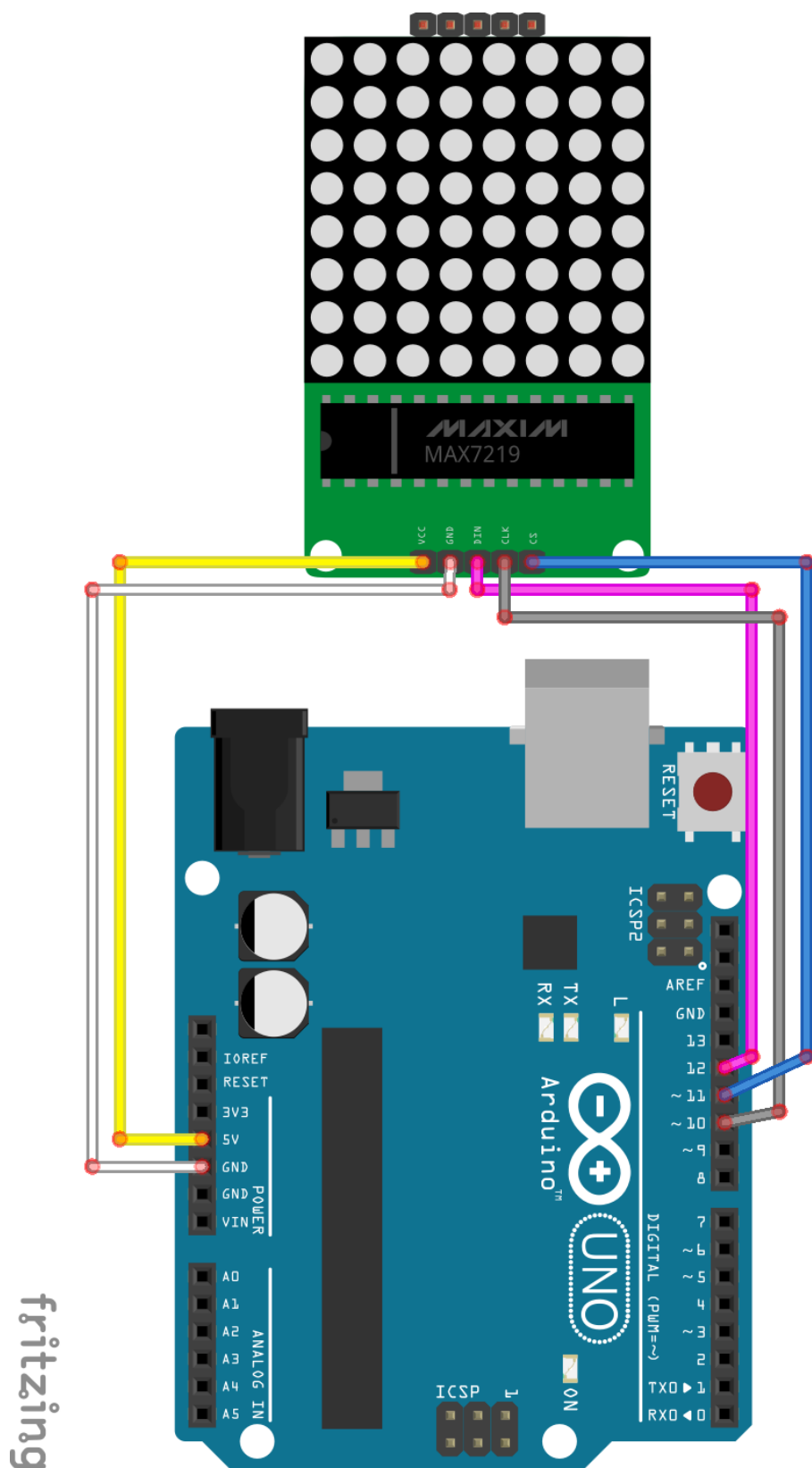


1.3 Joystick analogowy

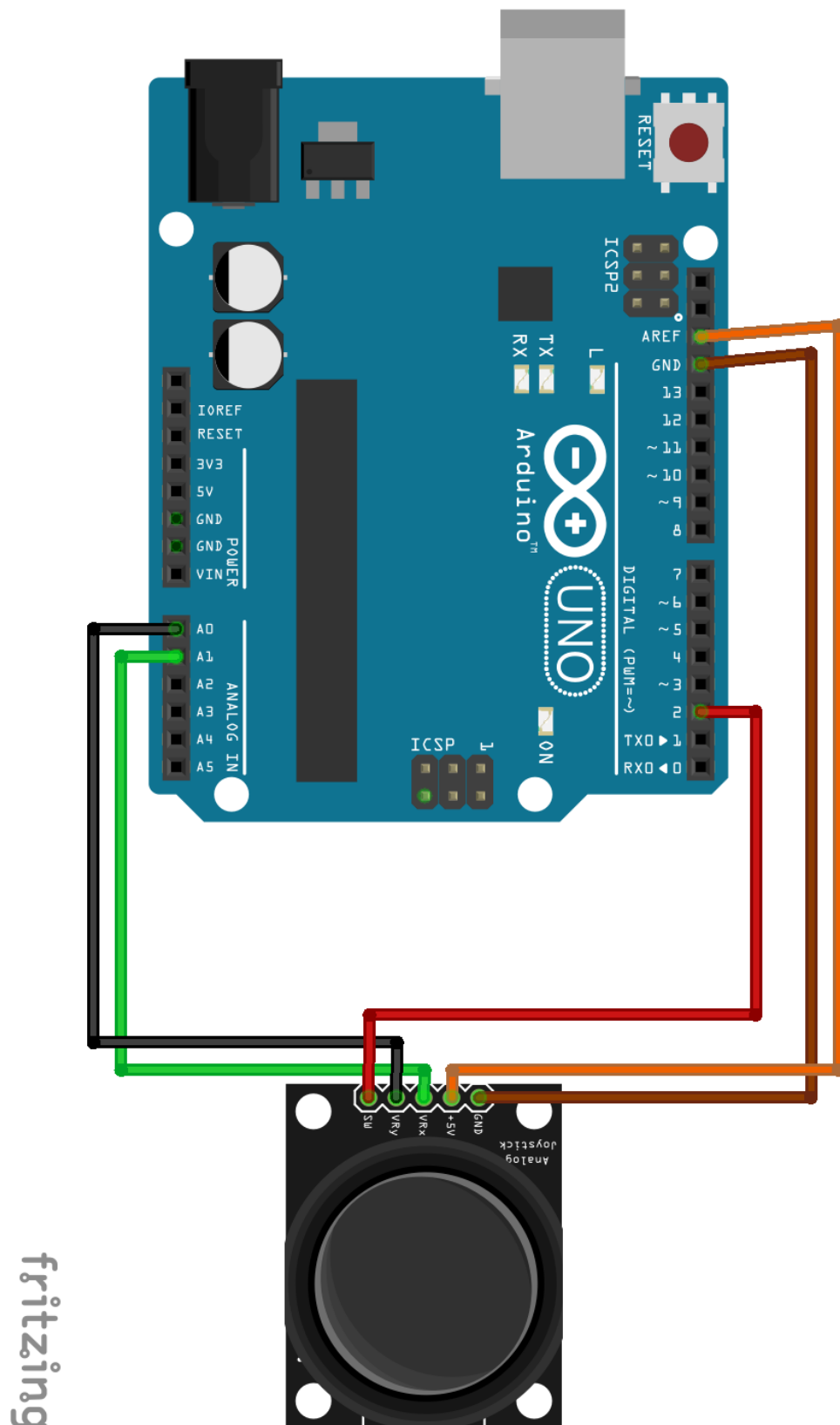


2 Schematy połączeń

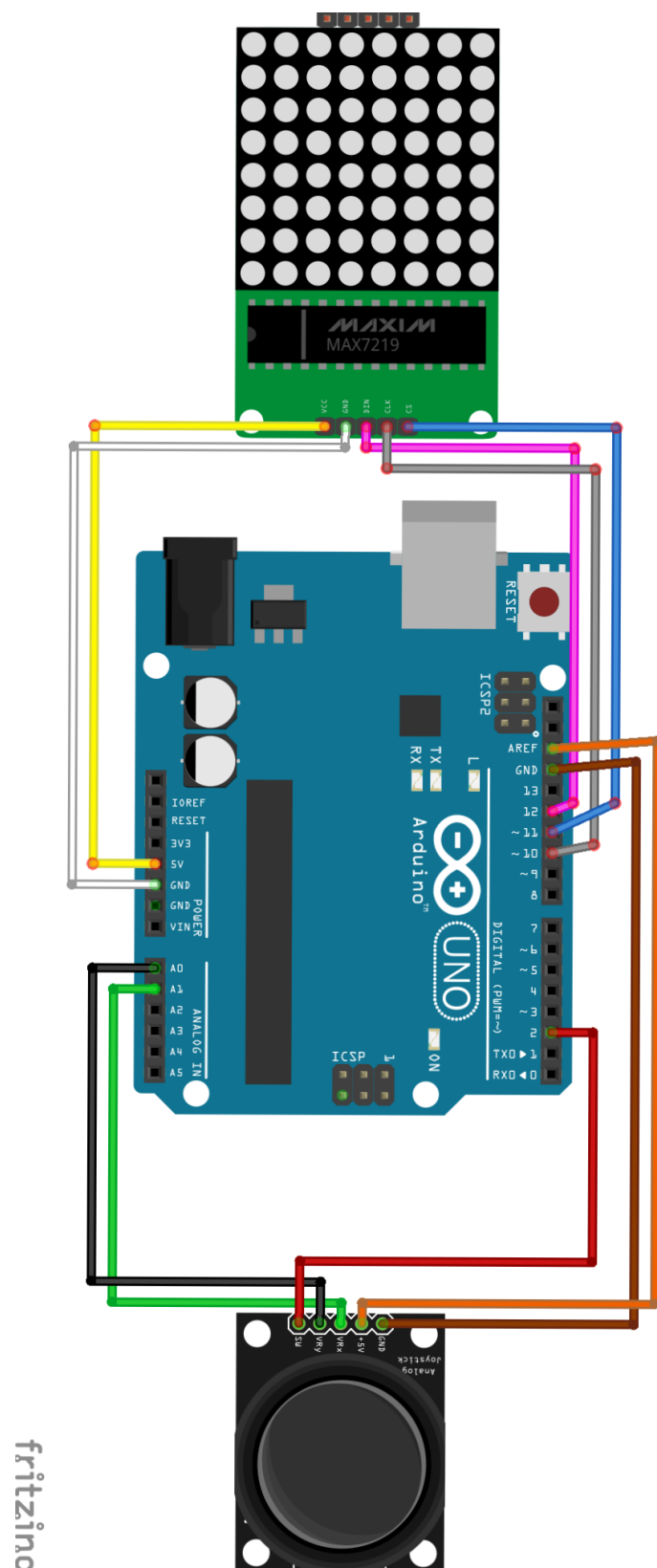
2.1 Arduino-Matryca LED



2.2 Arduino-Joystick



2.3 Całość



3 Opis programu

Gra rozpoczyna się widokiem rakiety, start nastąpi po naciśnięciu joysticka, wtedy piłeczka pojawi się losowo w jednym z dwóch pikseli. W tym celu wykorzystano funkcję *randomSeed* z argumentem *analogRead(A2)*, dzięki czemu funkcja *random* przy każdym uruchomieniu zwracała losową wartość, gdyż do wejścia *A2* nie podłączono żadnego sygnału. Na tym opiera się również losowość odbicia piłeczki od lewej krawędzi lub rakiety.

Pierwsze współrzędne pikseli rakiety są przechowywane w tablicy *bat*, a współrzędne piłeczki przechowują zmienne *r_ball* i *c_ball*. W trakcie gry funkcja *points* sprawdza czy piłeczka pokryła się z raketą i na tej podstawie są przyznawane punkty, które przechowuje zmienna *score1*. Punkty są wyświetlane w postaci binarnej na migającym ekranie (po przegranej) lub po naciśnięciu joysticka w trakcie gry w zerowej kolumnie. Wraz ze zdobyciem określonej ilości punktów, zwiększa się również poziom gry (zmienna *lvl*), a razem z nim tempo *SP_BALL* poruszania się piłeczki.

W trakcie gry jest możliwe przyspieszanie lub zwalnianie tempa poruszania się piłeczki. Ze względu na dużą czułość joysticka, łatwo było o przypadkową zmianę tempa. W tym celu zostało wprowadzone zabezpieczenie polegające na tym, że cały system zmiany tempa należy odblokować przez wychylenie joysticka w jedną ze stron (tak jak przy normalnej zmianie tempa), dopiero od tej chwili będą możliwe zmiany tempa w ciągu 800ms od poprzednich zmian (wychyleń). Tempa nie można zmienić powyżej wartości *SP_LIMIT* (im wyższa wartość *SP_BALL* tym piłeczka porusza się wolniej). Jeśli udało nam się przyspieszyć lub zwolnić tempo piłeczki zamigają dwie diody na górze lub dole ekranu, a w przypadku niepowodzenia tylko jedna.

Jeśli joystick zostanie naciśnięty na mniej niż 1 sekundę, w zerowej kolumnie pokażą się punkty, a w drugiej szanse. Co każdą sekundę przytrzymania joysticka nastąpi też zmiana rozmiaru rakiety. W momencie naciśnięcia joysticka gra w rzeczywistości się nie zatrzymuje, a bardzo zwalnia, piłeczka będzie niewidoczna i zmieniała pozycję co 30 sekund.

Aby piłeczka, rakietka oraz pozostałe mechanizmy mogły pracować niezależnie, pomocna okazała się funkcja *millis* umożliwiająca wprowadzenie wielozadaniowości. Do wyświetlania piłeczki, rakiety oraz innych elementów gry wykorzystana została funkcja *setLed*, którą zawiera biblioteka *LedControl*.

4 Biblioteka LedControl

Biblioteka *LedControl.h* jest biblioteką dla sterowników MAX7221 i MAX7219, które w prosty sposób umożliwiają kontrolę nad LED-ową matrycą 8x8 pikseli.

5 Listing kodu źródłowego

```
1
2 int RSIZE=1; // rozmiar paletki
3 int POINTS=3; // liczba żyć
4
5 int R = 0; int C = 0; //położenie joysticka R (row) i C (column)
6
7 int r_translate = 4; int c_translate = 7; // główny punkt rakiety (
  wszystkie inne są wyświetlane na podstawie tego, jeśli ustawimy RSIZE
  =1 to rakieta wyświetli się dokładnie w tym miejscu
8 int r_old = 4; // służy do czyszczenia starego położenia rakiety (stary
  wiersz), zmienna c_old nie jest potrzebna, bo rakieta zawsze jest w 7
  kolumnie
9
10 int c_ball = 0; //początkowo piłeczka znajduje się w 0 kolumnie
11 int r_ball;
12 int R_ball=1; // Przy rozpoczęciu gry, wiersz w którym pojawia się
  piłeczka jest losowy, ta zmienna kontroluje czy gra została wczytana
  poraz pierwszy
13 int br_old; int bc_old=0; //służą do czyszczenia starego położenia
  piłeczki ball row/column
14 int lr=2; //=1 piłeczka porusza się w lewo, =2 piłeczka porusza się w
  prawo
15
16 int SP_BALL=400; //prędkość piłeczki, w tej chwili piłeczka będzie
  zmieniała położenie co 400ms
17 int sp_controller1=1; int sp_controller2=0; //pomocnicze zmienne, które
  zostały wykorzystane do zabezpieczenia zmiany szybkości piłeczki, aby
  nie zmieniać jej przez przypadek w trakcie gry
18 unsigned long spch_timer = 0; //zmienna pomocnicza do zmiany tempa
  piłeczki, speed change
19 int led_sp=0; unsigned long led_sp_t=0; //gdy uda nam się zmienić tempo
  zaświeca się dwie diody (na gorze lub nadole w zależności od tego czy
  przyspieszamy czy zwalniamy)
20
21
22 int vec; // wykorzystane do tego aby piłeczka poruszała się w losowym
  kierunku
23 unsigned long now = 0; // https://forbot.pl/blog/kurs-arduino-ii-
  wielozadaniowosc-opoznienia-z-millis-id18418
24 unsigned long last = 0; //wykorzystane do szybkości przesuwania się
  rakiety
25 unsigned long b_last = 0; //wykorzystane do szybkości przesuwania się
  piłeczki
26 int bat[] = {-6,-6,-6,-6,-6}; // -6 ponieważ rakieta o rozmiarze 5 może
  się całkowicie znaleźć poza planszą i jeden punkt by miał wartość -5 (
  lub 5)
27
28 //Poziomy
29 int lvl=1; int score1=0; //aktualny lv i punkty
30 int SP_LIMIT=400; // najwolniejsze możliwe tempo piłeczki, nie będzie się
  dalo zwolnić powyżej tej wartości
31
32 //Rozmiar rakiety
```



```

33 int R_SIZE; //bedzie przechowywała rozmiar pileczki, który może się
    również zmieniać w trakcie gry
34 boolean firstgame=true; // wartość RSIZE będzie wykorzystana tylko przy
    pierwszym wczytaniu gry
35
36 //SW (switch czyli wciskanie joysticka)
37 unsigned long sw_timer = 0; //sprawdzi czy joy jest przytrzymywany
38 int sw_c=0; // służy do zmiany rozmiaru rakiety
39 int SP BALL_temp=SP BALL; //w momencie wcisnięcia joya gra bardzo zwalnia
40 int newgame=0; //służy do resetowania gry jeśli przegramy
41
42 #include "LedControl.h" // need the library
43 LedControl lc=LedControl(12,10,11,1); //10 is to CLOCK, 9 = CS, 8=DIN//
44 const int SW_pin = 2; //Switch do 2
45
46 void setup() {
47     Serial.begin(57600);
48     pinMode(SW_pin,INPUT);
49     digitalWrite(SW_pin, HIGH);
50
51     lc.shutdown(0,false); // turn off power saving, enables display
52     lc.setIntensity(0,0); // sets brightness (0~15 possible values)
53     lc.clearDisplay(0); // clear screen
54     randomSeed(analogRead(A2));
55 }
56
57 void loop() {
58     if (firstgame) R_SIZE=RSIZE;
59     if (POINTS>0){
60         //general
61         firstgame=false;
62         now = millis();
63         if (R_ball){r_ball = random(3,5); R_ball=0; br_old=r_ball;}
64
65         //rakieta
66         if (now - last >= 105){
67             last = now;
68             //czyszczenie rakiety
69             lc.setLed(0,r_old,c_translate,false);
70             if (R_SIZE>=2)lc.setLed(0,r_old-1,c_translate,false);
71             if (R_SIZE>=3)lc.setLed(0,r_old+1,c_translate,false);
72             if (R_SIZE>=4)lc.setLed(0,r_old-2,c_translate,false);
73             if (R_SIZE==5)lc.setLed(0,r_old+2,c_translate,false);
74
75             //Wyświetlanie i ruchy paletką
76             lc.setLed(0,r_translate,c_translate,true);
77             if (R_SIZE>=2)lc.setLed(0,r_translate-1,c_translate,true);
78             if (R_SIZE>=3)lc.setLed(0,r_translate+1,c_translate,true);
79             if (R_SIZE>=4)lc.setLed(0,r_translate-2,c_translate,true);
80             if (R_SIZE==5)lc.setLed(0,r_translate+2,c_translate,true);
81
82             if (R_SIZE>=4)bat[0]=r_translate-2;
83             if (R_SIZE>=2)bat[1]=r_translate-1;
84             bat[2]=r_translate;
85             if (R_SIZE>=3)bat[3]=r_translate+1;
86             if (R_SIZE==5)bat[4]=r_translate+2;

```

```

87     R = analogRead(A1);
88     int r_temp = map(R, 1023, 0, 7, 0);
89     r_old=r_translate; //do wyczyszczenia starego polozenia
90
91     //wartosci dla nowego polozenia rakietki
92     if(r_translate<7+R_SIZE and r_translate>0-R_SIZE){
93         if(r_temp>5) r_translate++; else if(r_temp<3) r_translate--;
94     }
95     else if(r_translate<=0){ if(r_temp>5) r_translate++;}
96     else {if(r_temp<3) r_translate--;}
97 }
98 if(newgame==0 and digitalRead(SW_pin)==HIGH)newgame=1;
99 if(newgame==1 and digitalRead(SW_pin)==LOW)newgame=2;
100 if(newgame==2){
101 //piłeczka
102 if(now - b_last >= SP_BALL){
103     b_last=now;
104     lc.setLed(0, br_old, bc_old, false);
105
106 //rysowanie piłeczki
107 lc.setLed(0, r_ball, c_ball, true);
108 br_old=r_ball; bc_old=c_ball;
109 //punkty
110 if(c_ball==7) {
111     if(points(r_ball, bat)==1){score1++;}
112     else{
113         POINTS-=1;
114         showPoints(POINTS);
115         showScore(score1);
116         delay(700);
117         lc.clearDisplay(0);
118     }
119 }
120 //wartość r dla nowego polozenia piłeczki
121 if(c_ball==0 and r_ball<=6 and r_ball>=1 and c_ball!=7){
122     vec=random(-1,2);
123     r_ball=vec+r_ball;
124 }
125 else if(r_ball==0){
126     r_ball++;
127     vec=1;
128 }
129 else if(r_ball==7){
130     r_ball--;
131     vec=-1;
132 }
133 else if(c_ball==7){
134     switch(random(0,5)){
135     case 0:
136         r_ball++;
137         break;
138     case 1:
139         r_ball--;
140         break;
141     case 2:
142         break;

```

```

143         default:
144             r_ball=r_ball+vec;
145         }
146     }
147     else{
148         r_ball=vec+r_ball;
149     }
150     //wartość c dla nowego położenia piłeczki
151     if(lr==2)c_ball++;
152     if(lr==1)c_ball--;
153     if(c_ball==0)lr=2;
154     if(c_ball==7)lr=1;
155 }
156 //Zmiana szybkości piłeczki
157 C = analogRead(A0);
158 int c_temp = map(C, 1023, 0, 0, 255);
159 int r_temp = map(R, 1023, 0, 255, 0);
160
161 if((c_temp==255 or c_temp==0) and r_temp==132 and
sp_controller1==1){sp_controller2=1; sp_controller1=0; spch_timer=now
; }
162 if(c_temp==127 and r_temp== 132 and sp_controller2==1){
sp_controller1=2;}
163 if((now-spch_timer>=50 and now-spch_timer<=800) and
sp_controller2==1 and sp_controller1==2){
164     //ZWOLNIENIE
165     if(c_temp==255 and r_temp== 132){
166         sp_controller2=0; sp_controller1=1;
167         if(SP_BALL<=SP_LIMIT-25){
168             SP_BALL+=25;
169             lc.setLed(0,7,3,true); lc.setLed(0,7,4,true); led_sp=1;
led_sp_t=now ;
170         }else{
171             SP_BALL=SP_LIMIT;
172             lc.setLed(0,7,3,true); led_sp=1; led_sp_t=now ;
173         }
174     }
175     //PRZYSPIESZENIE
176     else if(c_temp==0 and r_temp== 132){
177         sp_controller2=0; sp_controller1=1;
178         if(lvl<5){
179             SP_BALL-=25;
180             lc.setLed(0,0,3,true); lc.setLed(0,0,4,true); led_sp=1;
led_sp_t=now ;
181         }else{
182             SP_BALL=SP_LIMIT;
183             lc.setLed(0,0,3,true); led_sp=1; led_sp_t=now ;
184         }
185     }
186 }
187 //gaszenie ledów
188 if(led_sp==1 and now-led_sp_t>100){
189     led_sp=0;
190     lc.setLed(0,7,3,false); lc.setLed(0,7,4,false);
191     lc.setLed(0,0,3,false); lc.setLed(0,0,4,false);
192 }

```

```

193 //blokada zmiany tempa
194 else if (now-spch_timer>800){sp_controller1=1; sp_controller2
=0;}
195
196 //POZIOMY
197 if (score1==4){SP_LIMIT=360; if (SP BALL>360)SP BALL=360; lvl=2;}
198 if (score1==10){SP_LIMIT=320; if (SP BALL>320)SP BALL=320; lvl
=3;}
199 if (score1==20){SP_LIMIT=290; if (SP BALL>290)SP BALL=290; lvl=4;}
200 if (score1==30){SP_LIMIT=260; if (SP BALL>260)SP BALL=260; lvl=5;}
201 if (score1==40){SP_LIMIT=240; if (SP BALL>240)SP BALL=240; lvl=6;}
202 if (score1==50){SP_LIMIT=220; if (SP BALL>220)SP BALL=220; lvl=7;}
203 if (score1==60){SP_LIMIT=200; if (SP BALL>200)SP BALL=200; lvl=8;}
204
205 //Zmiana rozmiaru
206 if (digitalRead (SW_pin)==LOW) {
207     if (sw_c==0){
208         sw_c=1;
209         SP BALL_temp=SP BALL;
210         SP BALL=30000;
211         lc.setLed (0 , br_old , bc_old , false );
212     }
213     showScore (score1 );
214     showPoints (POINTS );
215     if (sw_c==1){sw_timer=now; sw_c=2;}
216     if (now-sw_timer>1000 and sw_c==2){
217         sw_c=1;
218         for (int i=0; i<=7;i++){
219             lc.setLed (0 , i , 7 , false );
220         }
221         if (R_SIZE==5){
222             R_SIZE=1;
223             bat [0] = -6; bat [1] = -6; bat [3] = -6; bat [4] = -6;
224         }
225         else R_SIZE++;
226     }
227 }
228 if (digitalRead (SW_pin)==HIGH and sw_c>=1){
229     for (int i=0; i<=7;i++){
230         lc.setLed (0 , i , 0 , false );
231     }
232     for (int i=0; i<=7;i++){
233         lc.setLed (0 , i , 2 , false );
234     }
235     sw_c=0;
236     SP BALL=SP BALL_temp;
237 }
238 }
239
240 }else if (POINTS==0){lc.clearDisplay (0);POINTS=-1; newgame=0;}
241 else{
242     now=millis ();
243
244     if (now-sw_timer<1500)showScore (score1 );
245     else if (now-sw_timer==1500) lc.clearDisplay (0);
246     else if (now-sw_timer>=1650) sw_timer=now;

```

```

247     if (digitalRead(SW_pin)==LOW){R = 0;lc.clearDisplay(0);
248         C = 0;
249         r_translate = 4;
250         c_translate = 7;
251         r_old = 4;bc_old=0;
252         c_ball = 0;lr=2;
253         SP_BALL=400; sp_controller1=1;sp_controller2=0;spch_timer = 0;
254         led_sp=0;led_sp_t=0;
255         R_ball=1;
256         now = 0;
257         last = 0;
258         b_last = 0;
259         bat[0]=-6; bat[1]=-6;bat[2]=-6;bat[3]=-6;bat[4]=-6;
260         //Poziomy
261         lvl=1;score1=0;
262         SP_LIMIT=400;
263         POINTS=3;
264         sw_timer = 0;
265         sw_c=0;
266         SP_BALL_temp=SP_BALL;
267         newgame=0;
268     }
269 }
270 }
271
272 //FUNKCJE
273 int points(int _r, int _bat[]){
274     for(int i=0; i<5; i++){
275         if(_r==_bat[i]) return 1;
276     }
277     return 0;
278 }
279
280 int showPoints(int _points){
281     int i=7;
282     for(; _points>0; _points--){
283         lc.setLed(0,7-i,2,true);
284         i--;
285     }
286 }
287
288 int showScore(int value){
289     int tab[8]={0,0,0,0,0,0,0,0};
290     for(int i=7; i>=0; i--){
291         if(value%2==1){
292             tab[i]=1;
293             value=(value-1)/2;
294         }
295         else{
296             tab[i]=0;
297             value=value/2;
298         }
299     }
300     for(int i=0; i<=7;i++){
301         if(tab[i]) lc.setLed(0,7-i,0,true);
302     }

```

