
PING PONG

Grupa numer 16
Informatyka
II rok
Rok akademicki 2017/2018

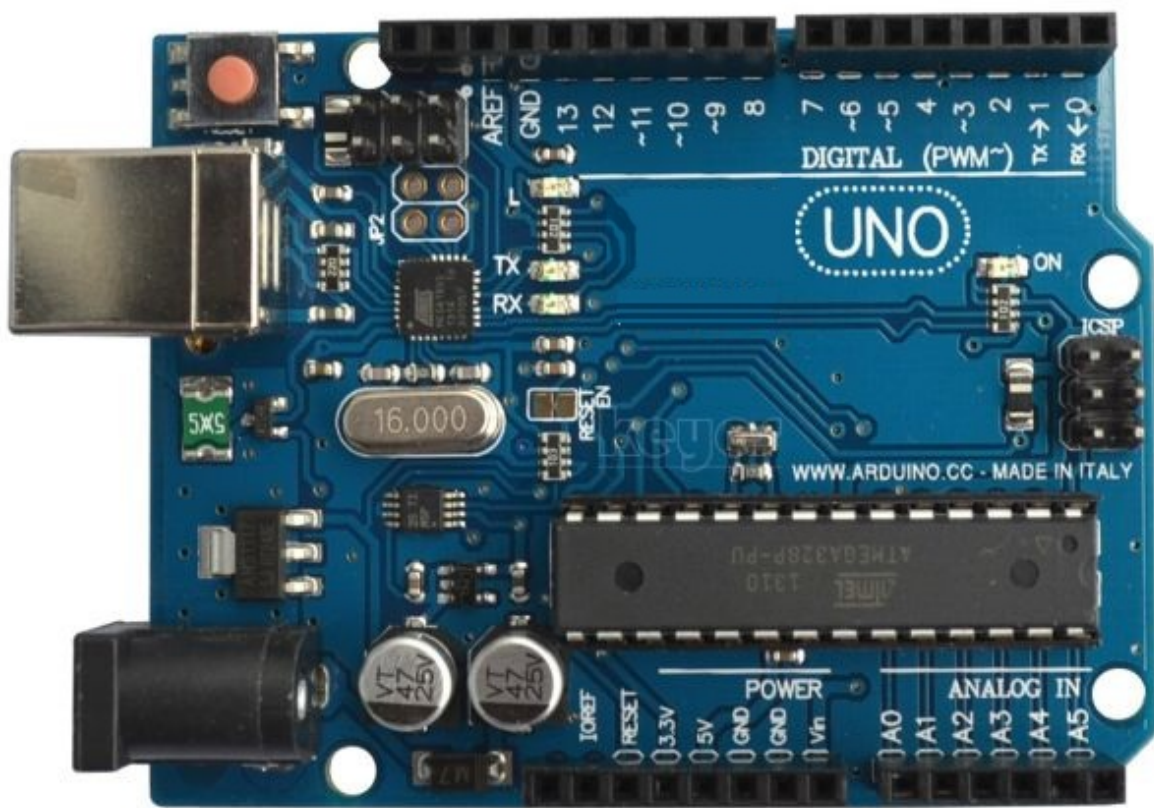
Autorzy:

KOLANO GRZEGORZ
WIATER MATEUSZ
WOLSKI ŁUKASZ

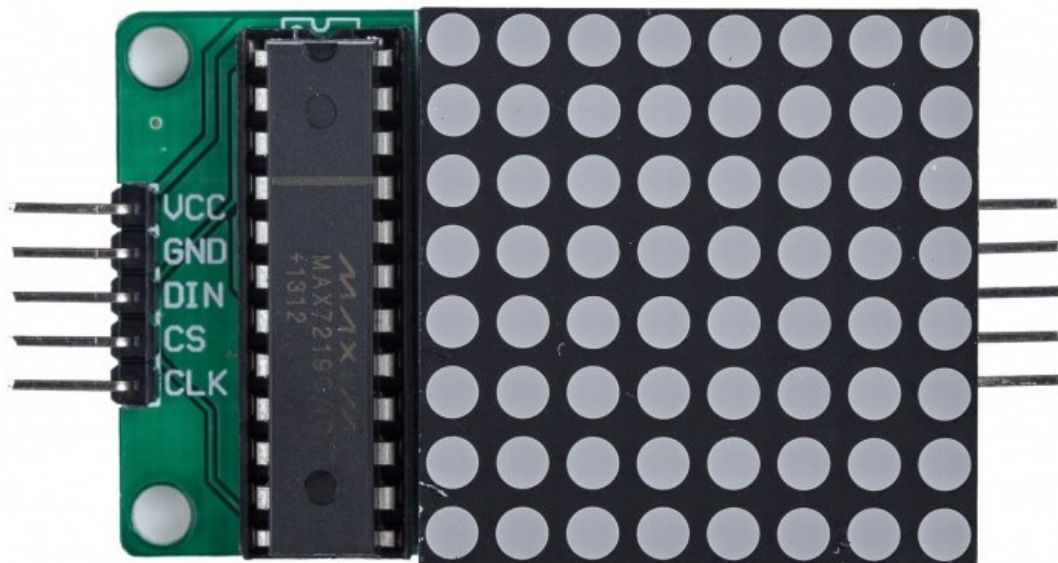
14 września 2018

1 Użyte moduły

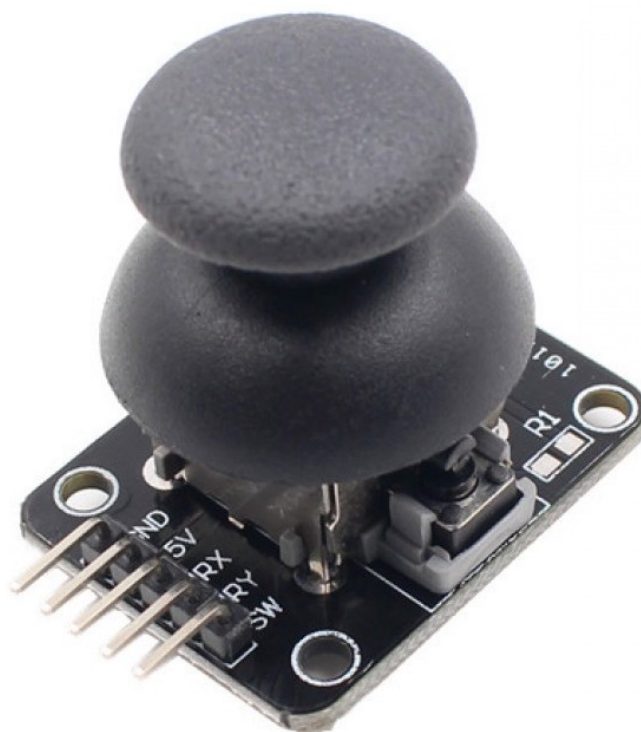
1.1 Arduino Uno



1.2 Matryca LED 8x8 ze sterownikiem MAX7219

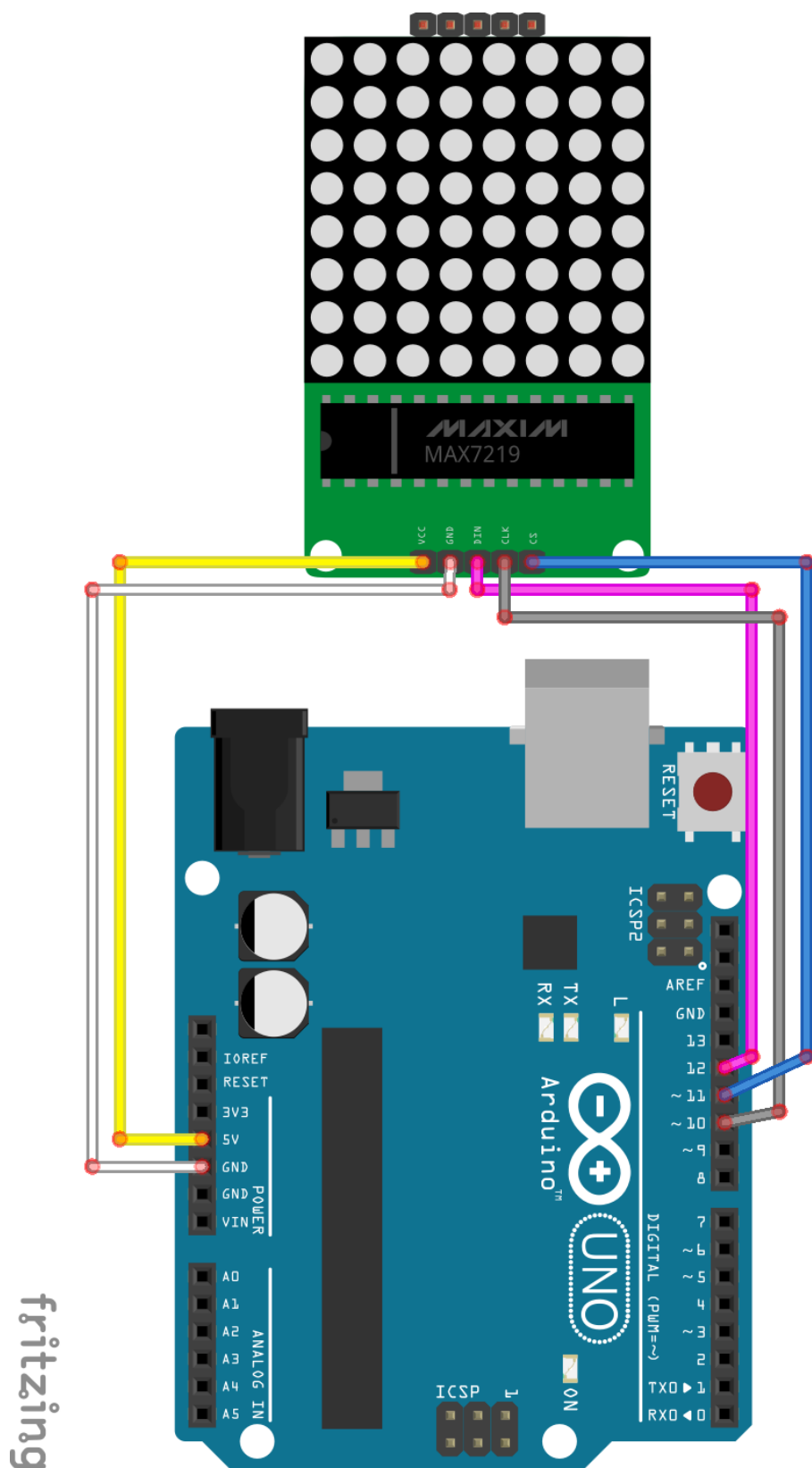


1.3 Joystick analogowy

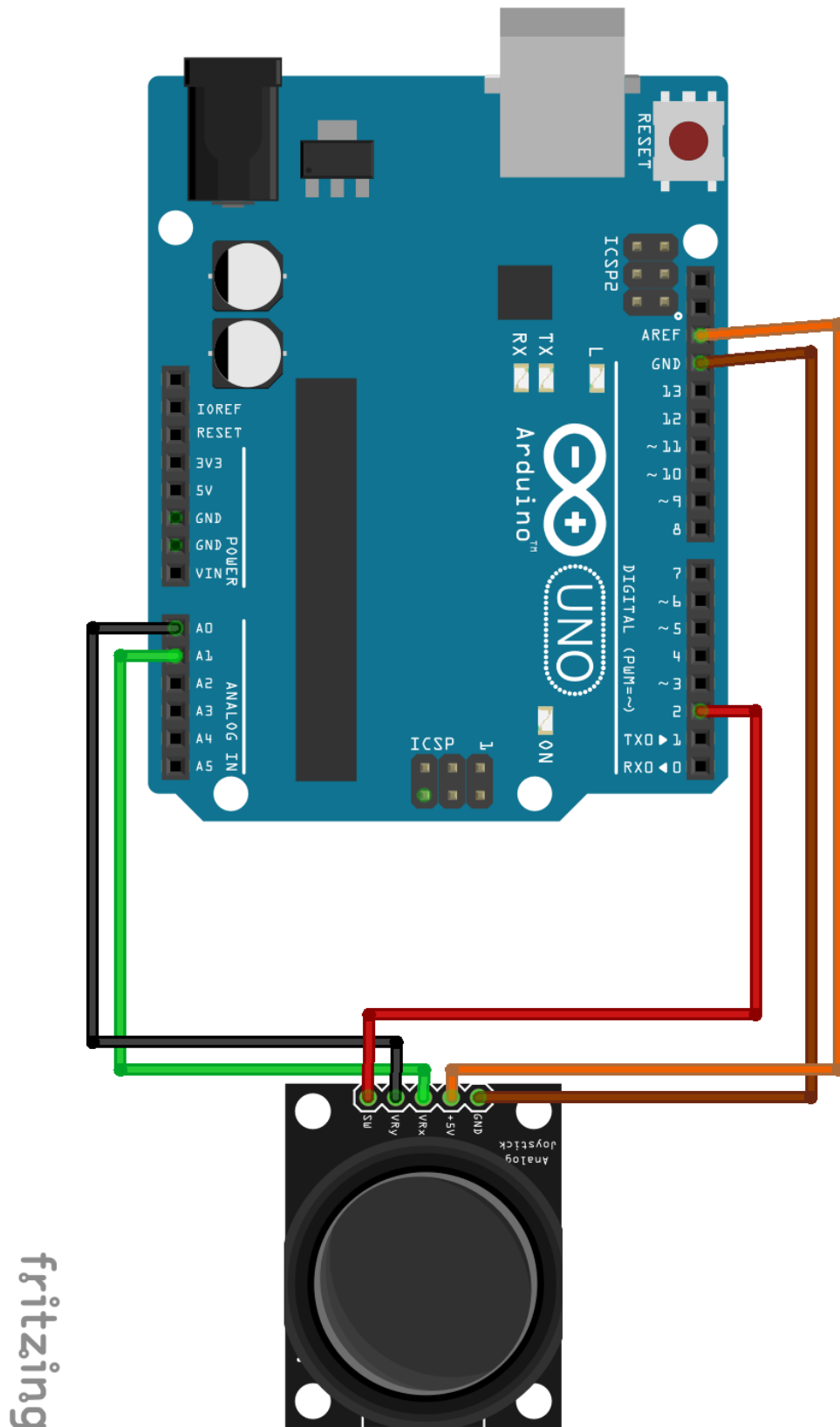


2 Schematy połączeń

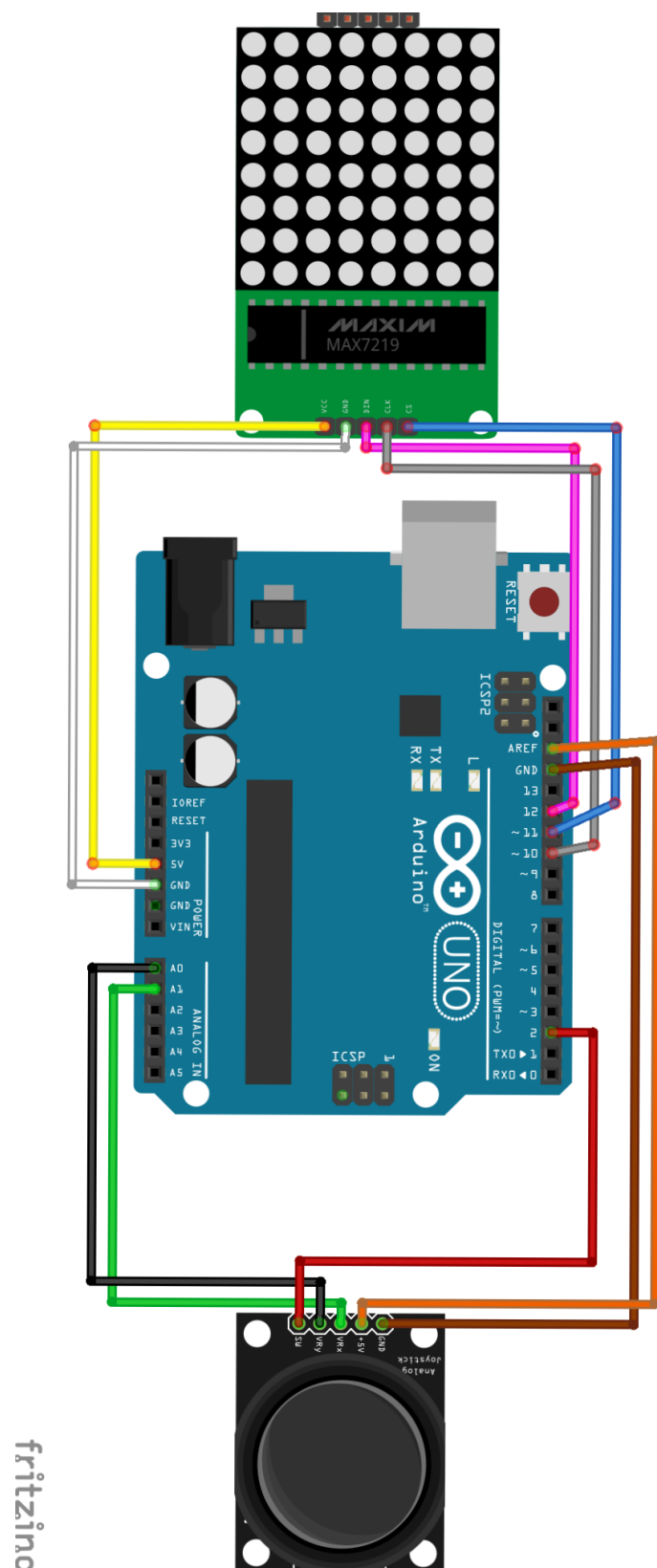
2.1 Arduino-Matryca LED



2.2 Arduino-Joystick



2.3 Całość



3 Opis działania 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=3; // rozmiar paletki
3 int POINTS=5; // liczba żyć
4
5 int R = 0; int C = 0; //położenie joysticka R (row) i C (column)
6 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
7 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
8 int c_ball = 0; //początkowo pileczka znajduje się w 0 kolumnie
9 int r_ball;
10 int R_ball=1; // Przy rozpoczęciu gry, wiersz w którym pojawi się
  pileczka jest losowy, ta zmienna kontroluje czy gra została wczytana
  poraz pierwszy
11 int br_old; int bc_old=0; //służą do czyszczenia starego położenia
  pileczki ball row/column
12 int lr=2; //1 pileczka porusza się w lewo, =2 pileczka porusza się w
  prawo
13 int SP_BALL=400; //predkosc pileczki, w tej chwili pileczka będzie
  zmieniała położenie co 400ms
14 int sp_controller1=1;int sp_controller2=0; //pomocnicze zmienne, które
  zostały wykorzystane do zabezpieczenia zmiany szybkości pileczki, aby
  nie zmieniać jej przez przypadek w trakcie gry
15 unsigned long spch_timer = 0; //zmienna pomocnicza do zmiany tempa
  pileczki, speed change
16 int led_sp=0; unsigned long led_sp_t=0; //gdy uda nam się zmienić tempo
  zaswieca się dwie diody (na gorze lub nadole w zależności od tego czy
  przyspieszamy czy zwalniamy)
17
18 int vec; // wykorzystane do tego aby pileczka poruszała się w losowym
  kierunku
19 unsigned long now = 0; // https://forbot.pl/blog/kurs-arduino-ii-
  wielozadaniowosc-opoznienia-z-millis-id18418
20 unsigned long last = 0; //wykorzystane do szybkości przesuwania się
  rakiety
21 unsigned long b_last = 0; //wykorzystane do szybkości przesuwania się
  pileczki
22 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)
23
24 //Poziomy
25 int lvl=1; int score1=0; //aktualny lv i punkty
26 int SP_LIMIT=400; // najwolniejsze możliwe tempo pileczki, nie będzie się
  dalo zwolnić powyżej tej wartości
27
28 //Rozmiar rakiety
29 int R_SIZE; //będzie przechowywała rozmiar pileczki, który może się
  również zmieniać w trakcie gry
30 boolean firstgame=true; // wartość RSIZE będzie wykorzystana tylko przy
  pierwszym wczytaniu gry
31
```



```

32 //SW (switch czyli wciskanie joysticka)
33 unsigned long sw_timer = 0; //sprawdzi czy joy jest przytrzymywany
34 int sw_c=0; // sluzy do zmiany rozmiaru rakietki
35 int SP BALL_temp=SP BALL; //w momencie wcisniecia joya gra bardzo zwalnia
36 int newgame=0; //sluzy do resetowania gry jesli przegramy
37
38 #include "LedControl.h" // need the library
39 LedControl lc=LedControl(12,10,11,1); //10 is to CLOCK, 9 = CS, 8=DIN//
40 const int SW_pin = 2; //Switch do 2
41
42 void setup() {
43     Serial.begin(57600);
44     pinMode(SW_pin,INPUT);
45     digitalWrite(SW_pin, HIGH);
46
47     lc.shutdown(0,false); // turn off power saving, enables display
48     lc.setIntensity(0,0); // sets brightness (0~15 possible values)
49     lc.clearDisplay(0); // clear screen
50     randomSeed(analogRead(A2));
51 }
52
53 void loop() {
54     if(firstgame)R_SIZE=RSIZE;
55     if(POINTS>0){
56         //general
57         firstgame=false;
58         now = millis();
59         if(R_ball){r_ball = random(3,5); R_ball=0; br_old=r_ball;}
60
61         //rakietka
62         if(now - last >= 105){
63             last = now;
64             //czyszczenie rakietki
65             lc.setLed(0,r_old,c_translate,false);
66             if(R_SIZE>=2)lc.setLed(0,r_old-1,c_translate,false);
67             if(R_SIZE>=3)lc.setLed(0,r_old+1,c_translate,false);
68             if(R_SIZE>=4)lc.setLed(0,r_old-2,c_translate,false);
69             if(R_SIZE==5)lc.setLed(0,r_old+2,c_translate,false);
70
71             //Wyświetlanie i ruchy paletką
72             lc.setLed(0,r_translate,c_translate,true);
73             if(R_SIZE>=2)lc.setLed(0,r_translate-1,c_translate,true);
74             if(R_SIZE>=3)lc.setLed(0,r_translate+1,c_translate,true);
75             if(R_SIZE>=4)lc.setLed(0,r_translate-2,c_translate,true);
76             if(R_SIZE==5)lc.setLed(0,r_translate+2,c_translate,true);
77
78             if(R_SIZE>=4)bat[0]=r_translate-2;
79             if(R_SIZE>=2)bat[1]=r_translate-1;
80             bat[2]=r_translate;
81             if(R_SIZE>=3)bat[3]=r_translate+1;
82             if(R_SIZE==5)bat[4]=r_translate+2;
83             R = analogRead(A1);
84             int r_temp = map(R, 1023, 0, 7, 0);
85             r_old=r_translate; //do wyczyszczenia starego polozenia
86
87

```

```

88 //wartosci dla nowego połozenia rakietki
89 if(r_translate<7+R_SIZE and r_translate>0-R_SIZE){
90     if(r_temp>5) r_translate++; else if(r_temp<3) r_translate--;
91 }
92 else if(r_translate<=0){ if(r_temp>5) r_translate++;}
93 else {if(r_temp<3) r_translate--;}
94 }
95 if(newgame==0 and digitalRead(SW_pin)==HIGH)newgame=1;
96 if(newgame==1 and digitalRead(SW_pin)==LOW)newgame=2;
97 if(newgame==2){
98 //piłeczka
99 if(now - b_last >= SP_BALL){
100     b_last=now;
101     lc.setLed(0, br_old, bc_old, false);
102
103 //rysowanie piłeczki
104 lc.setLed(0, r_ball, c_ball, true);
105 br_old=r_ball; bc_old=c_ball;
106 //punkty
107 if(c_ball==7) {
108     if(points(r_ball, bat)==1){score1++;}
109     else{ //dla straty punktu
110         POINTS-=1;
111         showPoints(POINTS);
112         showScore(score1);
113         delay(700);
114         lc.clearDisplay(0);
115     }
116 }
117 //wartość r dla nowego połozenia piłeczki
118 if(c_ball==0 and r_ball<=6 and r_ball>=1 and c_ball!=7){
119     vec=random(-1,2);
120     r_ball=vec+r_ball;
121 }
122 else if(r_ball==0){
123     r_ball++;
124     vec=1;
125 }
126 else if(r_ball==7){
127     r_ball--;
128     vec=-1;
129 }
130 else if(c_ball==7){
131     switch(random(0,5)){
132     case 0:
133         r_ball++;
134         break;
135     case 1:
136         r_ball--;
137         break;
138     case 2:
139         break;
140     default:
141         r_ball=r_ball+vec;
142     }
143 }

```

```

144         else{
145             r_ball=vec+r_ball;
146         }
147         //wartość c dla nowego położenia piłeczki
148         if(lr==2)c_ball++;
149         if(lr==1)c_ball--;
150         if(c_ball==0)lr=2;
151         if(c_ball==7)lr=1;
152     }
153     //Zmiana szybkości piłeczki
154     C = analogRead(A0);
155     int c_temp = map(C, 1023, 0, 0, 255);
156     int r_temp = map(R, 1023, 0, 255, 0);
157
158     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
; }
159     if(c_temp==127 and r_temp== 132 and sp_controller2==1){
sp_controller1=2;}
160     if((now-spch_timer>=50 and now-spch_timer<=800) and
sp_controller2==1 and sp_controller1==2){
161         //ZWOLNIENIE
162         if(c_temp==255 and r_temp== 132){
163             sp_controller2=0; sp_controller1=1;
164             if(SP_BALL<=SP_LIMIT-25){
165                 SP_BALL+=25;
166                 lc.setLed(0,7,3,true); lc.setLed(0,7,4,true); led_sp=1;
led_sp_t=now ;
167             } else {
168                 SP_BALL=SP_LIMIT;
169                 lc.setLed(0,7,3,true); led_sp=1; led_sp_t=now ;
170             }
171         }
172         //PRZYSPIESZENIE
173         else if(c_temp==0 and r_temp== 132){
174             sp_controller2=0; sp_controller1=1;
175             if(lvl<9){
176                 SP_BALL-=25;
177                 lc.setLed(0,0,3,true); lc.setLed(0,0,4,true); led_sp=1;
led_sp_t=now ;
178             } else {
179                 SP_BALL=SP_LIMIT;
180                 lc.setLed(0,0,3,true); led_sp=1; led_sp_t=now ;
181             }
182         }
183     }
184     //gaszenie ledów
185     if(led_sp==1 and now-led_sp_t>100){
186         led_sp=0;
187         lc.setLed(0,7,3,false); lc.setLed(0,7,4,false);
188         lc.setLed(0,0,3,false); lc.setLed(0,0,4,false);
189     }
190     //blokada zmiany tempa
191     else if(now-spch_timer>800){sp_controller1=1; sp_controller2
=0;}
192

```

```

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

```

```

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

```