

Ostatecznie do obsługi naszego bankomatu zostały napisane trzy programy:

- program główny,
- program do obsługi czytnika RFID,
- program do wykonywania nastaw.

Każdy z nich realizuje inne zadanie. Poniżej załączono krótki opis i zamieszczono kod każdego z programów.

## KOD PROGRAMU GŁÓWNEGO

Pierwszy z programów obsługuje całe działanie bankomatu ( obsługa: menu bankomatu, wyświetlacza, klawiatury i silników). W programie tym obsługa silników jest oparta na włączaniu przekaźnika sterującego na odpowiedni czas, jednakże wynika to z braku silnika wyposażonego w enkoder. Gdybyśmy posiadali taki właśnie silnik to wypłacanie banknotów oparte byłoby nie o czas obrotu silnika, a o jego kąt obrotu. Znając długość banknotu, a także obwód koła, które ma na celu wysunięcia banknotu bardzo łatwo obliczyć potrzebny kąt obrotu. Tym samym wystarczyłoby zliczać impulsy pochodzące z enkoder do odpowiedniego obrotu silnika.

Poniżej zamieszczam kod głównego programu

```
1 #include <EEPROM.h>
2 #include <LiquidCrystal.h>
3 #include <Keypad.h>
4
5
6 //      UART
7 // 48 - BRAK_KARTY
8 // 49 - KARTA_SERWISOWA
9 // 50 - KARTA_KLIEN1_1
10 // 51 - KARTA_KLIEN1_2
11
12 LiquidCrystal lcd(30, 31, 34, 35, 32, 33); //Wyświetlacz
13
14
15 const byte ROWS = 4; //Klawiatura
16 const byte COLS = 4;
17
18 byte rowPins[ROWS] = {23, 25, 27, 29};
19 byte colPins[COLS] = {22, 24, 26, 28};
20 char keys[ROWS][COLS] = {
21   {'1', '2', '3', '4'},
22   {'5', '6', '7', '8'},
23   {'9', '0', '*', '#'},
24   {'+', '-', '=', ' '}
25 };
26 Keypad klawiatura = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS ); //Zmienne
27 int ID;
28 int ID_start;
29 uint8_t stan;
30 uint8_t pole;
31 uint8_t liczba_prob;
32 uint8_t i;
33 char wprowadzony_PIN[4];
34 char PIN_serwisanta [4] = {'1', '1', '1', '1'};
35 char PIN_klienta_1[4] = {'0', '0', '0', '0'};
36 char PIN_klienta_2[4] = {'0', '0', '0', '0'};
37 uint8_t karta_serwisanta_blokada = EEPROM.read(6);
38 uint8_t karta_klienta_1_blokada = EEPROM.read(7);
39 uint8_t karta_klienta_2_blokada = EEPROM.read(8);
40 uint8_t az;
41 char klawisz;
42 int L_20 = EEPROM.read(9); //Liczba banknotów, kolejno 20z1, 50z1, 100z1
43 int L_50 = EEPROM.read(10);
44 int L_100 = EEPROM.read(11);
45 uint8_t stan_1;
46 uint8_t menu_serwisanta_1_s;
47 uint8_t menu_serwisanta_2_s;
48 int stan_konta_klienta_1 = ( (1000 * EEPROM.read(0)) + (100 * EEPROM.read(1)) + (10 * EEPROM.read(2)) ); //Stany kont klientów
49 int stan_konta_klienta_2 = ( (1000 * EEPROM.read(3)) + (100 * EEPROM.read(4)) + (10 * EEPROM.read(5)) );
50 int kwota_do_wplytu;
51 uint8_t L_20_wplyta = 0; //Liczba banknotów podczas wplytu
52 uint8_t L_50_wplyta = 0;
53 uint8_t L_100_wplyta = 0;
54 uint8_t wplyta = 0;
55
56
57 void wprowadzenie_PINu (uint8_t osoba = ID) //wprowadzanie PINu
58 {
59   Klawisz = klawiatura.getKey();
60
61   switch (pole) //wprowadzanie kolejnych pol PINu
62   {
63     case 0:
64       if (klawisz == '0' || klawisz == '1' || klawisz == '2' || klawisz == '3' || klawisz == '4' || klawisz == '5' || klawisz == '6' || klawisz == '7' || klawisz == '8' || klawisz == '9')
65       {
66         wprowadzony_PIN[0] = klawisz;
67         pole++;
68       }
69       break;
70
71     case 1:
72       if (klawisz == '0' || klawisz == '1' || klawisz == '2' || klawisz == '3' || klawisz == '4' || klawisz == '5' || klawisz == '6' || klawisz == '7' || klawisz == '8' || klawisz == '9')
73       {
74         wprowadzony_PIN[1] = klawisz;
75         pole++;
76       }
77       break;
78
79     case 2:
80       if (klawisz == '0' || klawisz == '1' || klawisz == '2' || klawisz == '3' || klawisz == '4' || klawisz == '5' || klawisz == '6' || klawisz == '7' || klawisz == '8' || klawisz == '9')
81       {
82         wprowadzony_PIN[2] = klawisz;
```

```

83     pole++;
84 }
85 break;
86
87 case 3:
88     if (klawisz == '0' || klawisz == '1' || klawisz == '2' || klawisz == '3' || klawisz == '4' || klawisz == '5' || klawisz == '6' || klawisz == '7' || klawisz == '8' || klawisz == '9')
89     {
90         wprowadzony_PIN[3] = klawisz;
91         pole++;
92     }
93     break;
94 }
95
96 if (pole == 4) //Sprawdzenie poprawnosci PINu
97 {
98     for (i : i < 4; i++)
99     {
100         if (osoba == 49 && (wprowadzony_PIN[i] == PIN_serwisanta[i]))
101         {
102             a++;
103         }
104
105         if (osoba == 50 && (wprowadzony_PIN[i] == PIN_klienta_1[i]))
106         {
107             a++;
108         }
109
110         if (osoba == 51 && (wprowadzony_PIN[i] == PIN_klienta_2[i]))
111         {
112             a++;
113         }
114     }
115     if (a == 4) //Poprawny PIN
116     {
117         lcd.clear();
118         lcd.setCursor(2, 0);
119         lcd.print("Poprawny PIN");
120         delay(1000);
121         pole = 0;
122         a = 0;
123         stan = 1;
124     }
125     else //Niepoprawny PIN
126     {
127         lcd.clear();
128         lcd.setCursor(0, 0);
129         lcd.print("Niepoprawny PIN");
130         liczba_prob--;
131         pole = 0;
132         a = 0;
133         i = 0;
134         wprowadzony_PIN[0] = '_';
135         wprowadzony_PIN[1] = '_';
136         wprowadzony_PIN[2] = '_';
137         wprowadzony_PIN[3] = '_';
138         delay(1000);
139     }
140 }
141
142 if (liczba_prob == 0) //Blokowanie karty
143 {
144     lcd.clear();
145     lcd.setCursor(5, 0);
146     lcd.print("KARTA");
147     lcd.setCursor(2, 2);
148     lcd.print("ZABLOKOWANA");
149
150     if (osoba == 49)
151     {
152         karta_serwisanta_blokada = 1;
153         EEPROM.update(6, 1);
154     }
155     if (osoba == 50)
156     {
157         karta_klienta_1_blokada = 1;
158         EEPROM.update(7, 1);
159     }
160     if (osoba == 51)
161     {
162         karta_klienta_2_blokada = 1;
163         EEPROM.update(8, 1);
164     }
165     delay(1000);
166 }
167 //koniec wprowadzania PINu
168
169 void menu_serwisanta_1()
170 {
171     if (stan_1 == 0)
172     {
173         stan_1 = 1;
174         menu_serwisanta_1_s = 1;
175     }
176
177     if (klawisz == 'D')
178     {
179         stan_1 = 0;
180         menu_serwisanta_1_s = 0;
181     }
182
183     if (klawisz == '1') //Zmiana liczby banknotow
184         L_20 ++;
185
186     if (klawisz == '2')
187         L_50++;
188
189     if (klawisz == '3')
190         L_100++;
191
192     if (klawisz == '7')
193         L_20 = L_20 + 5;
194
195     if (klawisz == '8')
196         L_50 = L_50 + 5;
197
198     if (klawisz == '9')
199         L_100 = L_100 + 5;
200
201
202
203
204
205

```

```

206
207 if (klawisz == '4' && L_20 > 0 )
208     L_20 --;
209
210 if (klawisz == '5' && L_50 > 0 )
211     L_50 --;
212
213 if (klawisz == '6' && L_100 > 0 )
214     L_100 --;
215
216 if (klawisz == '+' && L_20 > 4 )
217     L_20 = L_20 - 5;
218
219 if (klawisz == '0' && L_50 > 4 )
220     L_50 = L_50 - 5;
221
222 if (klawisz == '#' && L_100 > 4 )
223     L_100 = L_100 - 5;
224
225 if (klawisz == 'C')
226 {
227     lcd.clear();
228     lcd.setCursor(4, 0);
229     lcd.print("ZAPISANO"); //Zapisanie liczby banknotów
230     EEPROM.update(9, L_20);
231     EEPROM.update(10, L_50);
232     EEPROM.update(11, L_100);
233     delay(1000);
234 }
235
236
237
238 lcd.clear();
239 lcd.setCursor(0, 0);
240 lcd.print("20zl");
241 lcd.setCursor(6, 0);
242 lcd.print("50zl");
243 lcd.setCursor(11, 0);
244 lcd.print("100zl");
245 lcd.setCursor(0, 2);
246 lcd.print(L_20);
247 lcd.setCursor(6, 2);
248 lcd.print(L_50);
249 lcd.setCursor(11, 2);
250 lcd.print(L_100);
251
252 }
253
254 void menu_serwisanta_2() //Wybór karty do odblokowania
255 {
256     if (stan_1 == 0)
257     {
258         stan_1 = 1;
259         menu_serwisanta_2_s = 1;
260     }
261
262     if (klawisz == 'D')
263     {
264         stan_1 = 0;
265         menu_serwisanta_2_s = 0;
266     }
267
268     lcd.clear();
269     lcd.setCursor(0, 0);
270     lcd.print("Klient 1 [1]");
271     lcd.setCursor(0, 2);
272     lcd.print("Klient 2 [2]");
273
274     if (klawisz == '1')
275     {
276         if (karta_klienta_1_blokada == 1) //Odblokowywanie kart
277         {
278             lcd.clear();
279             lcd.setCursor(0, 0);
280             lcd.print("KARTA ZOSTALA");
281             lcd.setCursor(2, 2);
282             lcd.print("ODBLOKOWANA");
283             karta_klienta_1_blokada = 0;
284             EEPROM.update(7, 0);
285             delay(1000);
286         }
287         else
288         {
289             lcd.clear();
290             lcd.setCursor(0, 0);
291             lcd.print("KARTA JEST");
292             lcd.setCursor(2, 2);
293             lcd.print("ODBLOKOWANA");
294             delay(1000);
295         }
296     }
297
298     if (klawisz == '2')
299     {
300         if (karta_klienta_2_blokada == 1)
301         {
302             lcd.clear();
303             lcd.setCursor(0, 0);
304             lcd.print("KARTA ZOSTALA");
305             lcd.setCursor(2, 2);
306             lcd.print("ODBLOKOWANA");
307             karta_klienta_2_blokada = 0;
308             EEPROM.update(8, 0);
309             delay(1000);
310         }
311         else
312         {
313             lcd.clear();
314             lcd.setCursor(0, 0);
315             lcd.print("KARTA JEST");
316             lcd.setCursor(2, 2);
317             lcd.print("ODBLOKOWANA");
318             delay(1000);
319         }
320     }
321 }
322
323 void stan_konta(uint8_t osoba_stan_konta = ID) //Stan konta
324 {
325     lcd.clear();
326     lcd.setCursor(0, 0);
327     lcd.print("Twoj stan konta");
328

```

```

329 if (osoba_stan_konta == 50)
330 {
331   lcd.setCursor(6, 2);
332   lcd.print(stan_konta_klienta_1);
333   lcd.setCursor(14, 2);
334   lcd.print("z1");
335 }
336
337 if (osoba_stan_konta == 51)
338 {
339   lcd.setCursor(6, 2);
340   lcd.print(stan_konta_klienta_2);
341   lcd.setCursor(14, 2);
342   lcd.print("z1");
343 }
344
345 if (klawisz == 'D')
346   stan = 1;
347
348 }
349 menu_wypлата(uint8_t osoba_wypлата = ID) //Menu wypлата
350 {
351   if (klawisz == 'D')
352     stan = 1;
353
354   if (klawisz == '1' && (kwota_do_wypłaty + 20) <= 500)
355     kwota_do_wypłaty = kwota_do_wypłaty + 20;
356
357   if (klawisz == '2' && (kwota_do_wypłaty + 50) <= 500)
358     kwota_do_wypłaty = kwota_do_wypłaty + 50;
359
360   if (klawisz == '3' && (kwota_do_wypłaty + 100) <= 500)
361     kwota_do_wypłaty = kwota_do_wypłaty + 100;
362
363   if (klawisz == '4' && (kwota_do_wypłaty - 20) >= 0)
364     kwota_do_wypłaty = kwota_do_wypłaty - 20;
365
366   if (klawisz == '5' && (kwota_do_wypłaty - 50) >= 0)
367     kwota_do_wypłaty = kwota_do_wypłaty - 50;
368
369   if (klawisz == '6' && (kwota_do_wypłaty - 100) >= 0)
370     kwota_do_wypłaty = kwota_do_wypłaty - 100;
371
372   lcd.clear();
373   lcd.setCursor(0, 0);
374   lcd.print("Ile wypłacasz?");
375   lcd.setCursor(6, 2);
376   lcd.print(kwota_do_wypłaty);
377   lcd.setCursor(10, 2);
378   lcd.print("z1");
379
380   if (osoba_wypлата == 50 && klawisz == 'C' && kwota_do_wypłaty != 0) //Wypлата Klient 1
381   {
382
383     if (kwota_do_wypłaty <= stan_konta_klienta_1)
384     {
385       L_100_wypлата = (kwota_do_wypłaty / 100);
386
387       if (L_100_wypлата <= L_100)
388       {
389         L_50_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата)) / 50);
390
391         if (L_50_wypлата <= L_50)
392           L_20_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата) - (50 * L_50_wypлата)) / 20);
393
394         if (L_50_wypлата > L_50)
395         {
396           L_50_wypлата = L_50;
397           L_20_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата) - (50 * L_50_wypлата)) / 20);
398         }
399       }
400
401       if (L_100_wypлата > L_100)
402       {
403         L_100_wypлата = L_100;
404         L_50_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата)) / 50);
405
406         if (L_50_wypлата <= L_50)
407           L_20_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата) - (50 * L_50_wypлата)) / 20);
408
409         if (L_50_wypлата > L_50)
410         {
411           L_50_wypлата = L_50;
412           L_20_wypлата = ((kwota_do_wypłaty - (100 * L_100_wypлата) - (50 * L_50_wypлата)) / 20);
413         }
414       }
415     }
416
417     if ( ( kwota_do_wypłaty - (L_100_wypлата * 100) - (L_50_wypлата * 50) - (L_20_wypлата * 20) ) == 0)
418     {
419       lcd.clear();
420       lcd.setCursor(2, 0);
421       lcd.print("REALIZOWANIE");
422       stan_konta_klienta_1 = stan_konta_klienta_1 - kwota_do_wypłaty;
423       L_20 = L_20 - L_20_wypлата;
424       L_50 = L_50 - L_50_wypлата;
425       L_100 = L_100 - L_100_wypлата;
426       EEPROM.update(0, (stan_konta_klienta_1 / 1000)); //Zapis stanu konta klienta_1
427       EEPROM.update(1, ((stan_konta_klienta_1 - (EEPROM.read(0) * 1000)) / 100));
428       EEPROM.update(2, ((stan_konta_klienta_1 - (EEPROM.read(0) * 1000) - (EEPROM.read(1) * 100)) / 10));
429       EEPROM.update(9, L_20); //Zapis, ile banknotów pozostało w bankomacie
430       EEPROM.update(10, L_50);
431       EEPROM.update(11, L_100);
432       kwota_do_wypłaty = 0;
433
434       for ( L_100_wypлата; L_100_wypлата > 0; L_100_wypлата --) //Obsługa siłników podczas wypłaty
435       {
436         digitalWrite(36, LOW);
437         digitalWrite(38, HIGH);
438         digitalWrite(39, HIGH);
439         delay(1000);
440         digitalWrite(36, HIGH);
441         digitalWrite(38, LOW);
442         digitalWrite(39, LOW);
443         delay(500);
444       }
445
446       for ( L_50_wypлата; L_50_wypлата > 0; L_50_wypлата --)
447       {
448         digitalWrite(36, LOW);
449         digitalWrite(38, HIGH);
450
451

```

```

452     delay(1000);
453     digitalWrite(36, HIGH);
454     digitalWrite(39, LOW);
455     delay(500);
456 }
457
458 for ( L_20_wyplata; L_20_wyplata > 0; L_20_wyplata --)
459 {
460     digitalWrite(36, LOW);
461     delay(1000);
462     digitalWrite(36, HIGH);
463     delay(500);
464 }
465
466 wyplata = 1;
467 stan = 1;
468
469 }
470 else
471 {
472     lcd.clear(); //Brak wyplaty - nie ma odpowiednich banknotow
473     lcd.setCursor(4, 0);
474     lcd.print("WYPLATA");
475     lcd.setCursor(3, 2);
476     lcd.print("NIEMOZLIWA");
477     kwota_do_wyplaty = 0;
478     delay(1000);
479 }
480 }
481 else
482 {
483     lcd.clear(); //Brak wyplaty - za malo srodkow na koncie
484     lcd.setCursor(0, 0);
485     lcd.print("BRAK SRODKOW");
486     lcd.setCursor(2, 2);
487     lcd.print("NA KONCIE");
488     kwota_do_wyplaty = 0;
489     delay(2000);
490 }
491 }
492
493 if (osoba_wyplata == 51 && klawisz == 'C' && kwota_do_wyplaty != 0) //Wyplata klient 2
494 {
495
496     if (kwota_do_wyplaty <= stan_konta_klienta_2)
497     {
498         L_100_wyplata = (kwota_do_wyplaty / 100);
499
500         if (L_100_wyplata <= L_100)
501         {
502             L_50_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata)) / 50);
503
504             if (L_50_wyplata <= L_50)
505                 L_20_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata) - (50 * L_50_wyplata)) / 20);
506
507             if (L_50_wyplata > L_50)
508             {
509                 L_50_wyplata = L_50;
510                 L_20_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata) - (50 * L_50_wyplata)) / 20);
511             }
512         }
513
514         if (L_100_wyplata > L_100)
515         {
516             L_100_wyplata = L_100;
517             L_50_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata)) / 50);
518
519             if (L_50_wyplata <= L_50)
520                 L_20_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata) - (50 * L_50_wyplata)) / 20);
521
522             if (L_50_wyplata > L_50)
523             {
524                 L_50_wyplata = L_50;
525                 L_20_wyplata = ((kwota_do_wyplaty - (100 * L_100_wyplata) - (50 * L_50_wyplata)) / 20);
526             }
527         }
528     }
529 }
530
531 if (( kwota_do_wyplaty - (L_100_wyplata * 100) - (L_50_wyplata * 50) - (L_20_wyplata * 20) == 0)
532 {
533     lcd.clear();
534     lcd.setCursor(2, 0);
535     lcd.print("REALIZOWANIE");
536     stan_konta_klienta_2 = stan_konta_klienta_2 - kwota_do_wyplaty;
537     L_20 = L_20 - L_20_wyplata;
538     L_50 = L_50 - L_50_wyplata;
539     L_100 = L_100 - L_100_wyplata;
540     EEPROM.update(3, (stan_konta_klienta_1 / 1000)); //Zapis stanu konta klienta_2
541     EEPROM.update(4, ((stan_konta_klienta_1 - (EEPROM.read(3) * 1000)) / 100));
542     EEPROM.update(5, ((stan_konta_klienta_1 - (EEPROM.read(3) * 1000) - (EEPROM.read(4) * 100)) / 10)); //Zapis, ile banknotow pozostalo w bankomacie
543     EEPROM.update(9, L_20);
544     EEPROM.update(10, L_50);
545     EEPROM.update(11, L_100);
546     kwota_do_wyplaty = 0;
547     for ( L_100_wyplata; L_100_wyplata > 0; L_100_wyplata --)
548     {
549         digitalWrite(36, LOW);
550         digitalWrite(38, HIGH);
551         digitalWrite(39, HIGH);
552         delay(1000);
553         digitalWrite(36, HIGH);
554         digitalWrite(38, LOW);
555         digitalWrite(39, LOW);
556         delay(500);
557     }
558
559     for ( L_50_wyplata; L_50_wyplata > 0; L_50_wyplata --)
560     {
561         digitalWrite(36, LOW);
562         digitalWrite(39, HIGH);
563         delay(1000);
564         digitalWrite(36, HIGH);
565         digitalWrite(39, LOW);
566         delay(500);
567     }
568
569     for ( L_20_wyplata; L_20_wyplata > 0; L_20_wyplata --)
570     {
571         digitalWrite(36, LOW);
572     }
573
574 }

```

```

575     delay(1000);
576     digitalWrite(36, HIGH);
577     delay(500);
578 }
579 wypлата = 1;
580 stan = 1;
581 }
582 }
583 else
584 {
585     lcd.clear(); //Brak wypłaty - nie ma odpowiednich banknotow
586     lcd.setCursor(4, 0);
587     lcd.print("WYPLATA");
588     lcd.setCursor(3, 2);
589     lcd.print("NIEMOZLIWA");
590     kwota_do_wypłaty = 0;
591     delay(1000);
592 }
593 }
594 else
595 {
596     lcd.clear(); //Brak wypłaty - za mało srodkow na koncie
597     lcd.setCursor(0, 0);
598     lcd.print(" BRAK SRODKOW");
599     lcd.setCursor(2, 2);
600     lcd.print(" NA KONCIE");
601     kwota_do_wypłaty = 0;
602     delay(1000);
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 void setup() {
612     stan = 0;
613     stan_1 = 0;
614     pole = 0;
615     a = 0;
616     i = 0;
617     liczba_prob = 3;
618     menu_serwisanta_1_s = 0;
619     menu_serwisanta_2_s = 0;
620     Serial3.begin(9600);
621     lcd.begin(16, 2);
622     lcd.clear();
623     lcd.setCursor(2, 0);
624     lcd.print("Dzien dobry!");
625     lcd.setCursor(2, 2);
626     lcd.print("Wloz karte!");
627     wprowadzony_PIN[0] = '-';
628     wprowadzony_PIN[1] = '-';
629     wprowadzony_PIN[2] = '-';
630     wprowadzony_PIN[3] = '-';
631     kwota_do_wypłaty = 0;
632     wypłata = 0;
633     pinMode(36, OUTPUT);
634     pinMode(38, OUTPUT);
635     pinMode(39, OUTPUT);
636     digitalWrite(36, HIGH);
637 }
638 }
639 }
640 }
641 void loop() {
642     ID = Serial3.read();
643     if (ID != ID_stare) //Zmniejszenie częstotliwości odświeżania (wykrywanie zmiany ID)
644     {
645         switch (ID) //switch wyboru osoby
646         {
647             case 48: //Brak karty
648                 stan = 0;
649                 stan_1 = 0;
650                 pole = 0;
651                 a = 0;
652                 i = 0;
653                 liczba_prob = 3;
654                 menu_serwisanta_1_s = 0;
655                 menu_serwisanta_2_s = 0;
656                 wprowadzony_PIN[0] = '-';
657                 wprowadzony_PIN[1] = '-';
658                 wprowadzony_PIN[2] = '-';
659                 wprowadzony_PIN[3] = '-';
660                 kwota_do_wypłaty = 0;
661                 wypłata = 0;
662                 lcd.clear();
663                 lcd.setCursor(2, 0);
664                 lcd.print("Dzien dobry!");
665                 lcd.setCursor(2, 2);
666                 lcd.print("Wloz karte!");
667                 break;
668             case 49: //Karta serwisanta
669                 if (karta_serwisanta_blokada == 0)
670                 {
671                     switch (stan)
672                     {
673                         {
674                             case 0:
675                                 lcd.clear();
676                                 lcd.setCursor(0, 0);
677                                 lcd.print("Podaj PIN");
678                                 lcd.setCursor(0, 2);
679                                 lcd.print("Liczba prob");
680                                 lcd.setCursor(12, 0);
681                                 lcd.print(wprowadzony_PIN[0]);
682                                 lcd.setCursor(13, 0);
683                                 lcd.print(wprowadzony_PIN[1]);
684                                 lcd.setCursor(14, 0);
685                                 lcd.print(wprowadzony_PIN[2]);
686                                 lcd.setCursor(15, 0);
687                                 lcd.print(wprowadzony_PIN[3]);
688                                 lcd.setCursor(12, 2);
689                                 lcd.print(liczba_prob);
690                                 wprowadzenie_PINu (ID);
691                                 break;
692                             case 1:
693                                 if (stan_1 == 0)
694                                 {
695                                     lcd.clear();
696                                 }
697                             }
675     delay(1000);
676     digitalWrite(36, HIGH);
677     delay(500);
678 }
679 wypłata = 1;
680 stan = 1;
681 }
682 }
683 else
684 {
685     lcd.clear(); //Brak wypłaty - nie ma odpowiednich banknotow
686     lcd.setCursor(4, 0);
687     lcd.print("WYPLATA");
688     lcd.setCursor(3, 2);
689     lcd.print("NIEMOZLIWA");
690     kwota_do_wypłaty = 0;
691     delay(1000);
692 }
693 }
694 else
695 {
696     lcd.clear(); //Brak wypłaty - za mało srodkow na koncie
697     lcd.setCursor(0, 0);
698     lcd.print(" BRAK SRODKOW");
699     lcd.setCursor(2, 2);
700     lcd.print(" NA KONCIE");
701     kwota_do_wypłaty = 0;
702     delay(1000);
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 void setup() {
712     stan = 0;
713     stan_1 = 0;
714     pole = 0;
715     a = 0;
716     i = 0;
717     liczba_prob = 3;
718     menu_serwisanta_1_s = 0;
719     menu_serwisanta_2_s = 0;
720     Serial3.begin(9600);
721     lcd.begin(16, 2);
722     lcd.clear();
723     lcd.setCursor(2, 0);
724     lcd.print("Dzien dobry!");
725     lcd.setCursor(2, 2);
726     lcd.print("Wloz karte!");
727     wprowadzony_PIN[0] = '-';
728     wprowadzony_PIN[1] = '-';
729     wprowadzony_PIN[2] = '-';
730     wprowadzony_PIN[3] = '-';
731     kwota_do_wypłaty = 0;
732     wypłata = 0;
733     pinMode(36, OUTPUT);
734     pinMode(38, OUTPUT);
735     pinMode(39, OUTPUT);
736     digitalWrite(36, HIGH);
737 }
738 }
739 }
740 }
741 void loop() {
742     ID = Serial3.read();
743     if (ID != ID_stare) //Zmniejszenie częstotliwości odświeżania (wykrywanie zmiany ID)
744     {
745         switch (ID) //switch wyboru osoby
746         {
747             case 48: //Brak karty
748                 stan = 0;
749                 stan_1 = 0;
750                 pole = 0;
751                 a = 0;
752                 i = 0;
753                 liczba_prob = 3;
754                 menu_serwisanta_1_s = 0;
755                 menu_serwisanta_2_s = 0;
756                 wprowadzony_PIN[0] = '-';
757                 wprowadzony_PIN[1] = '-';
758                 wprowadzony_PIN[2] = '-';
759                 wprowadzony_PIN[3] = '-';
760                 kwota_do_wypłaty = 0;
761                 wypłata = 0;
762                 lcd.clear();
763                 lcd.setCursor(2, 0);
764                 lcd.print("Dzien dobry!");
765                 lcd.setCursor(2, 2);
766                 lcd.print("Wloz karte!");
767                 break;
768             case 49: //Karta serwisanta
769                 if (karta_serwisanta_blokada == 0)
770                 {
771                     switch (stan)
772                     {
773                         {
774                             case 0:
775                                 lcd.clear();
776                                 lcd.setCursor(0, 0);
777                                 lcd.print("Podaj PIN");
778                                 lcd.setCursor(0, 2);
779                                 lcd.print("Liczba prob");
780                                 lcd.setCursor(12, 0);
781                                 lcd.print(wprowadzony_PIN[0]);
782                                 lcd.setCursor(13, 0);
783                                 lcd.print(wprowadzony_PIN[1]);
784                                 lcd.setCursor(14, 0);
785                                 lcd.print(wprowadzony_PIN[2]);
786                                 lcd.setCursor(15, 0);
787                                 lcd.print(wprowadzony_PIN[3]);
788                                 lcd.setCursor(12, 2);
789                                 lcd.print(liczba_prob);
790                                 wprowadzenie_PINu (ID);
791                                 break;
792                             case 1:
793                                 if (stan_1 == 0)
794                                 {
795                                     lcd.clear();
796                                 }
797                             }
775     delay(1000);
776     digitalWrite(36, HIGH);
777     delay(500);
778 }
779 wypłata = 1;
780 stan = 1;
781 }
782 }
783 else
784 {
785     lcd.clear(); //Brak wypłaty - nie ma odpowiednich banknotow
786     lcd.setCursor(4, 0);
787     lcd.print("WYPLATA");
788     lcd.setCursor(3, 2);
789     lcd.print("NIEMOZLIWA");
790     kwota_do_wypłaty = 0;
791     delay(1000);
792 }
793 }
794 else
795 {
796     lcd.clear(); //Brak wypłaty - za mało srodkow na koncie
797     lcd.setCursor(0, 0);
798     lcd.print(" BRAK SRODKOW");
799     lcd.setCursor(2, 2);
800     lcd.print(" NA KONCIE");
801     kwota_do_wypłaty = 0;
802     delay(1000);
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 void setup() {
812     stan = 0;
813     stan_1 = 0;
814     pole = 0;
815     a = 0;
816     i = 0;
817     liczba_prob = 3;
818     menu_serwisanta_1_s = 0;
819     menu_serwisanta_2_s = 0;
820     Serial3.begin(9600);
821     lcd.begin(16, 2);
822     lcd.clear();
823     lcd.setCursor(2, 0);
824     lcd.print("Dzien dobry!");
825     lcd.setCursor(2, 2);
826     lcd.print("Wloz karte!");
827     wprowadzony_PIN[0] = '-';
828     wprowadzony_PIN[1] = '-';
829     wprowadzony_PIN[2] = '-';
830     wprowadzony_PIN[3] = '-';
831     kwota_do_wypłaty = 0;
832     wypłata = 0;
833     pinMode(36, OUTPUT);
834     pinMode(38, OUTPUT);
835     pinMode(39, OUTPUT);
836     digitalWrite(36, HIGH);
837 }
838 }
839 }
840 }
841 void loop() {
842     ID = Serial3.read();
843     if (ID != ID_stare) //Zmniejszenie częstotliwości odświeżania (wykrywanie zmiany ID)
844     {
845         switch (ID) //switch wyboru osoby
846         {
847             case 48: //Brak karty
848                 stan = 0;
849                 stan_1 = 0;
850                 pole = 0;
851                 a = 0;
852                 i = 0;
853                 liczba_prob = 3;
854                 menu_serwisanta_1_s = 0;
855                 menu_serwisanta_2_s = 0;
856                 wprowadzony_PIN[0] = '-';
857                 wprowadzony_PIN[1] = '-';
858                 wprowadzony_PIN[2] = '-';
859                 wprowadzony_PIN[3] = '-';
860                 kwota_do_wypłaty = 0;
861                 wypłata = 0;
862                 lcd.clear();
863                 lcd.setCursor(2, 0);
864                 lcd.print("Dzien dobry!");
865                 lcd.setCursor(2, 2);
866                 lcd.print("Wloz karte!");
867                 break;
868             case 49: //Karta serwisanta
869                 if (karta_serwisanta_blokada == 0)
870                 {
871                     switch (stan)
872                     {
873                         {
874                             case 0:
875                                 lcd.clear();
876                                 lcd.setCursor(0, 0);
877                                 lcd.print("Podaj PIN");
878                                 lcd.setCursor(0, 2);
879                                 lcd.print("Liczba prob");
880                                 lcd.setCursor(12, 0);
881                                 lcd.print(wprowadzony_PIN[0]);
882                                 lcd.setCursor(13, 0);
883                                 lcd.print(wprowadzony_PIN[1]);
884                                 lcd.setCursor(14, 0);
885                                 lcd.print(wprowadzony_PIN[2]);
886                                 lcd.setCursor(15, 0);
887                                 lcd.print(wprowadzony_PIN[3]);
888                                 lcd.setCursor(12, 2);
889                                 lcd.print(liczba_prob);
890                                 wprowadzenie_PINu (ID);
891                                 break;
892                             case 1:
893                                 if (stan_1 == 0)
894                                 {
895                                     lcd.clear();
896                                 }
897                             }
875     delay(1000);
876     digitalWrite(36, HIGH);
877     delay(500);
878 }
879 wypłata = 1;
880 stan = 1;
881 }
882 }
883 else
884 {
885     lcd.clear(); //Brak wypłaty - nie ma odpowiednich banknotow
886     lcd.setCursor(4, 0);
887     lcd.print("WYPLATA");
888     lcd.setCursor(3, 2);
889     lcd.print("NIEMOZLIWA");
890     kwota_do_wypłaty = 0;
891     delay(1000);
892 }
893 }
894 else
895 {
896     lcd.clear(); //Brak wypłaty - za mało srodkow na koncie
897     lcd.setCursor(0, 0);
898     lcd.print(" BRAK SRODKOW");
899     lcd.setCursor(2, 2);
900     lcd.print(" NA KONCIE");
901     kwota_do_wypłaty = 0;
902     delay(1000);
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }

```

```

698     lcd.setCursor(0, 0);
699     lcd.print("L.banknotow [A]");
700     lcd.setCursor(0, 2);
701     lcd.print("Odb. kart [B]");
702 }
703
704     klawisz = klawiatura.getKey();
705     if (klawisz == 'A' || menu_serwisanta_1_s)
706     {
707         menu_serwisanta_1();
708     }
709
710     if (klawisz == 'B' || menu_serwisanta_2_s)
711     {
712         menu_serwisanta_2();
713     }
714
715     break;
716 }
717 }
718 else
719 {
720     lcd.clear();
721     lcd.setCursor(5, 0);
722     lcd.print("KARTA");
723     lcd.setCursor(2, 2);
724     lcd.print("ZABLOKOWANA");
725 }
726 break;
727
728 case 50: //Karta klienta_1
729
730
731
732     if (karta_klienta_1_blokada == 0)
733     {
734         switch (stan)
735         {
736             case 0:
737                 lcd.clear();
738                 lcd.setCursor(0, 0);
739                 lcd.print("Podaj PIN");
740                 lcd.setCursor(0, 2);
741                 lcd.print("Liczba prob");
742                 lcd.setCursor(12, 0);
743                 lcd.print(wprowadzony_PIN[0]);
744                 lcd.setCursor(13, 0);
745                 lcd.print(wprowadzony_PIN[1]);
746                 lcd.setCursor(14, 0);
747                 lcd.print(wprowadzony_PIN[2]);
748                 lcd.setCursor(15, 0);
749                 lcd.print(wprowadzony_PIN[3]);
750                 lcd.setCursor(12, 2);
751                 lcd.print(liczba_prob);
752                 wprowadzenie_PINu (ID);
753                 break;
754
755             case 1:
756                 klawisz = klawiatura.getKey();
757                 lcd.clear();
758                 lcd.setCursor(0, 0);
759                 lcd.print("Stan konta [A]");
760                 lcd.setCursor(0, 2);
761                 lcd.print("Wypłata [B]");
762
763                 if (klawisz == 'A')
764                     stan = 2;
765
766                 if (klawisz == 'B' && wypłata == 0)
767                 {
768                     stan = 3;
769                 }
770
771                 if (klawisz == 'B' && wypłata == 1)
772                 {
773                     lcd.clear();
774                     lcd.setCursor(1, 0);
775                     lcd.print("WŁOZ POWNONIE");
776                     lcd.setCursor(5, 2);
777                     lcd.print("KARTE");
778                     delay(1000);
779                 }
780                 break;
781
782             case 2:
783                 klawisz = klawiatura.getKey();
784                 stan_konta(ID);
785                 break;
786
787             case 3:
788                 klawisz = klawiatura.getKey();
789                 menu_wypłata(ID);
790                 break;
791         }
792     }
793 }
794 else
795 {
796     lcd.clear();
797     lcd.setCursor(5, 0);
798     lcd.print("KARTA");
799     lcd.setCursor(2, 2);
800     lcd.print("ZABLOKOWANA");
801 }
802 break;
803
804 case 51: //Karta klienta_2
805     if (karta_klienta_2_blokada == 0)
806     {
807         switch (stan)
808         {
809             case 0:
810                 lcd.clear();
811                 lcd.setCursor(0, 0);
812                 lcd.print("Podaj PIN");
813                 lcd.setCursor(0, 2);
814                 lcd.print("Liczba prob");
815                 lcd.setCursor(12, 0);
816                 lcd.print(wprowadzony_PIN[0]);
817                 lcd.setCursor(13, 0);
818                 lcd.print(wprowadzony_PIN[1]);
819                 lcd.setCursor(14, 0);
820

```

```

821     lcd.print(wprowadzony_PIN[2]);
822     lcd.setCursor(15, 0);
823     lcd.print(wprowadzony_PIN[3]);
824     lcd.setCursor(12, 2);
825     lcd.print(liczba_prob);
826     wprowadzenie_PINu (ID);
827     break;
828
829     case 1:
830         klawisz = klawiatura.getKey();
831         lcd.clear();
832         lcd.setCursor(0, 0);
833         lcd.print("Stan konta [A]");
834         lcd.setCursor(0, 2);
835         lcd.print("Wypłata [B]");
836
837
838         if (klawisz == 'A')
839             stan = 2;
840
841         if (klawisz == 'B' && wypłata == 0)
842         {
843             stan = 3;
844         }
845         if (klawisz == 'B' && wypłata == 1)
846         {
847             lcd.clear();
848             lcd.setCursor(1, 0);
849             lcd.print("WŁOZ PONIOWIE");
850             lcd.setCursor(5, 2);
851             lcd.print("KARTĘ");
852             delay(1000);
853         }
854         break;
855
856     case 2:
857         klawisz = klawiatura.getKey();
858         stan_konta(ID);
859         break;
860
861     case 3:
862         klawisz = klawiatura.getKey();
863         menu_wypłata(ID);
864         break;
865
866     }
867 }
868 else
869 {
870     lcd.clear();
871     lcd.setCursor(5, 0);
872     lcd.print("KARTA");
873     lcd.setCursor(2, 2);
874     lcd.print("ZABLOKOWANA");
875 }
876 }
877
878 break;
879 } //koniec switcha od wyboru osoby
880
881 }
882
883 ID_stare = ID;
884
885 ;
886
887

```



## KOD PROGRAMU DO OBSŁUGI CZYTNIKA RFID

Program ten ma na celu wykrywanie, która z kart RFID jest przyłożona do bankomatu. Gdy karta zostanie wykryta to informację o niej są przesyłane z Arduino Nano do Arduino MEGA 2560 przy pomocy portu szeregowego (UART). Tak naprawdę kod ten został napisany tylko z powodu tego, że posiadane przez nas Arduino MEGA 2560 ma uszkodzona magistralę SPI, w związku z tym nie może ono obsługiwać czytnika kart RFID. W przypadku gdybyśmy posiadali w pełni sprawne Arduino MEGA 2560, ten kod byłby po prostu częścią kodu głównego.

```
1 /* Podłączenie
2 *
3 * -----
4 * MFRCS22 Arduino Arduino Arduino Arduino Arduino
5 * Reader/PCD Uno/101 Mega Nano v3 Leonardo/Micro Pro Micro
6 * Signal Pin Pin Pin Pin Pin Pin Pin
7 *
8 * RST/Reset RST 9 5 D9 RESET/ICSP-5 RST
9 * SPI SS SDA(SS) 10 53 D10 10 10
10 * SPI MOSI MOSI 11 / ICSP-4 51 D11 ICSP-4 16
11 * SPI MISO MISO 12 / ICSP-1 50 D12 ICSP-1 14
12 * SPI SCK SCK 13 / ICSP-3 52 D13 ICSP-3 15
13 */
14 #include <SPI.h>
15 #include <MFRCS22.h>
16
17 #define SS_PIN 10
18 #define RST_PIN 9
19
20
21 MFRCS22 rfid(SS_PIN, RST_PIN); //Deklaracja PINów do obsługi czytnika
22
23 MFRCS22::MIFARE_Key key;
24
25 const byte karta_serwis[] = { 192, 94, 253, 50}; //Numery ID kart
26 const byte karta_klient_1[] = { 208, 206, 14, 50};
27 const byte karta_klient_2[] = { 176, 40, 86, 50};
28 int a = 0;
29 int karta;
30
31 void setup() {
32   Serial.begin(9600);
33   SPI.begin(); //Inicjacja magistrali SPI
34   rfid.PCD_Init(); //Inicjacja czytnika RFID
35
36 }
37
38
39 void loop() {
40   if(rfid.PICC_IsNewCardPresent()==0) //Wykrywanie braku karty z zabezpieczeniem przed zakłóceniami
41   {
42     a++;
43     if(a>5)
44     {
45       a=5;
46       karta=0;
47     }
48   }
49   else
50   {
51     a=0;
52     karta=1;
53   }
54 }
55 rfid.PICC_ReadCardSerial();
56
57
58 if(karta == 1 && rfid.uid.uidByte[0]==192 && rfid.uid.uidByte[1]==94 && rfid.uid.uidByte[2]==253 && rfid.uid.uidByte[3]==50) //Sprawdzenie, która z kart jest obecna
59   Serial.print(1);
60
61 if(karta == 1 && rfid.uid.uidByte[0]==208 && rfid.uid.uidByte[1]==206 && rfid.uid.uidByte[2]==14 && rfid.uid.uidByte[3]==50)
62   Serial.print(2);
63
64 if(karta == 1 && rfid.uid.uidByte[0]==176 && rfid.uid.uidByte[1]==40 && rfid.uid.uidByte[2]==86 && rfid.uid.uidByte[3]==50)
65   Serial.print(3);
66
67 if(karta == 0)
68   Serial.print(0);
69
70   delay(500);
71
72
73   rfid.PCD_StopCryptol();
74
75
76 }
```

## KOD PROGRAMU DO WYKONYWANIA NASTAW

Program ten umożliwia zapis do pamięci EEPROM odpowiednich nastaw. Umożliwia on między innymi zmianę parametrów, które można zmodyfikować za pomocą menu serwisanta ( modyfikacja liczby banknotów jaka znajdują się w bankomacie, odblokowywanie kart klientów), ale także tych których modyfikacja nie jest możliwa z poziomu menu serwisanta (odblokowywanie karty serwisanta, modyfikacja stanu kont klientów).

```
1 #include <EEPROM.h>
2
3
4
5
6
7 uint8_t klient_1_1000zl = 1;           //Stan konta klienta_1
8 uint8_t klient_1_100zl = 2;
9 uint8_t klient_1_10zl = 4;
10 uint8_t klient_2_1000zl = 5;         //Stan konta klienta_2
11 uint8_t klient_2_100zl = 5;
12 uint8_t klient_2_10zl = 8;
13 uint8_t karta_serwisanta_blokada = 0; //Stan blokad kart
14 uint8_t karta_klienta_1_blokada = 0;
15 uint8_t karta_klienta_2_blokada = 0;
16 uint8_t L_20 = 5;                    //Liczba banknotów w bankomacie
17 uint8_t L_50 = 5;
18 int L_100 = 5;
19
20
21 void setup()
22 {
23   EEPROM.update(0, klient_1_1000zl); //Zapis zmiennych do pamięci EEPROM
24   EEPROM.update(1, klient_1_100zl);
25   EEPROM.update(2, klient_1_10zl);
26   EEPROM.update(3, klient_2_1000zl);
27   EEPROM.update(4, klient_2_100zl);
28   EEPROM.update(5, klient_2_10zl);
29   EEPROM.update(6, karta_serwisanta_blokada);
30   EEPROM.update(7, karta_klienta_1_blokada);
31   EEPROM.update(8, karta_klienta_2_blokada);
32   EEPROM.update(9, L_20);
33   EEPROM.update(10, L_50);
34   EEPROM.update(11, L_100);
35 }
36
37 void loop()
38 {
39
40 }
```

Napisany przez nas kod jest napisany do obsługi Arduino, zgodnie z tym co zostało przez nas napisane w pierwszym raporcie. Nasz program łatwo można wykorzystać do obsługi sterownika LEGO. Zgodnie z informacjami znalezionymi w zasobach sieciowych, w bardzo łatwy sposób można połączyć Arduino z sterownikiem LEGO przy pomocy magistrali I2C, dlatego też w modelu CAD naszego bankomatu oprócz Arduino jest także sterownik LEGO. Według naszego zamysłu Arduino MEGA 2560 miałyby obsługiwać czytnik RFID, wyświetlacz oraz klawiaturę, a sterownik LEGO miałby za zadanie obsługiwać silniki. Arduino MEGA 2560 przechowywałoby również dane niezbędne do funkcjonowania bankomatu, w pamięci EEPROM.