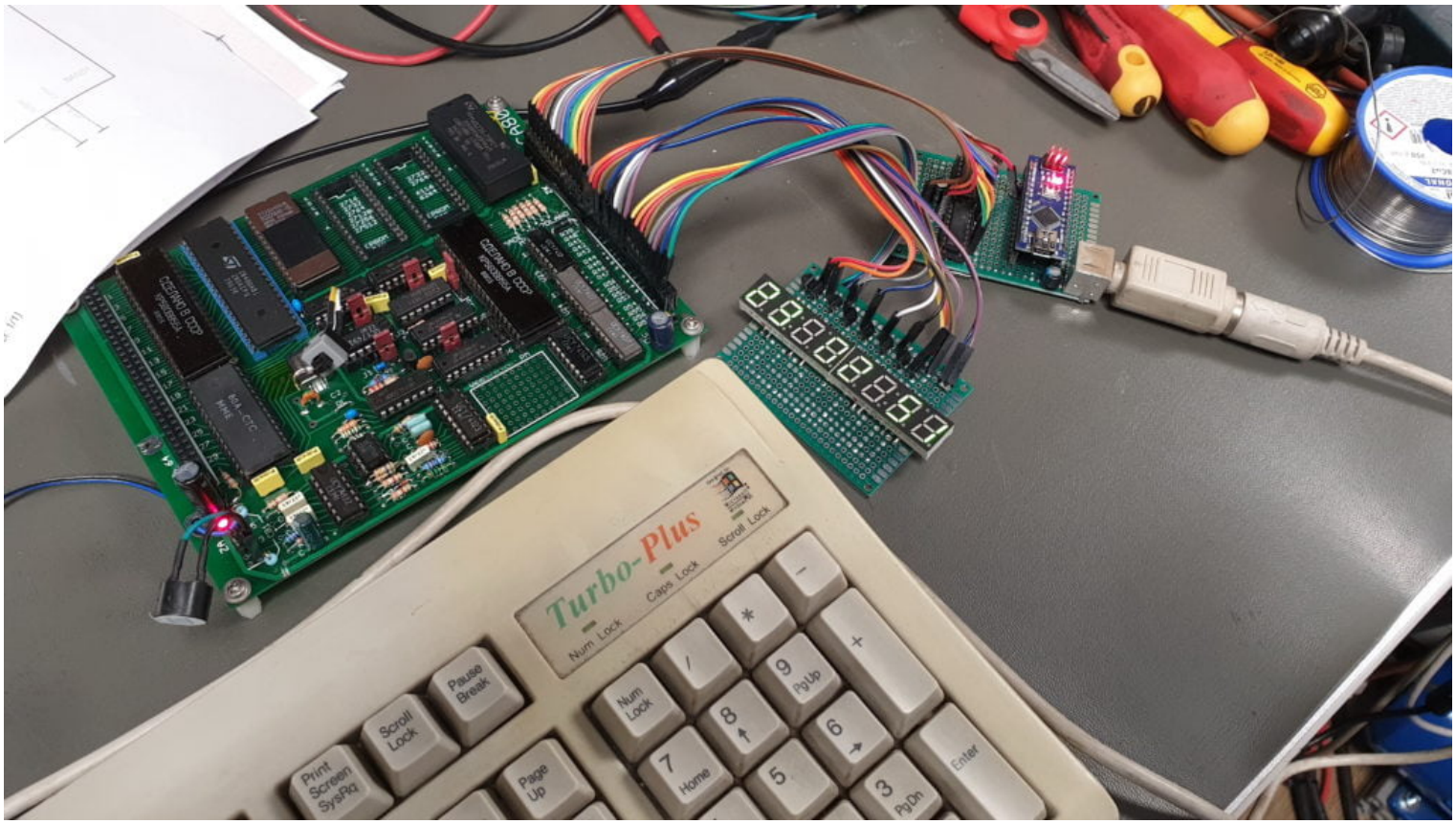
Projekt interfejsu komunikacyjnego umożliwiającego współpracę klawiatury PC posiadającej interfejs PS/2 z mikrokomputerem retro CA80.

Autor: Sławomir Jurkiewicz



Mikrokomputer CA80 z interfejsem i klawiaturą PS/2

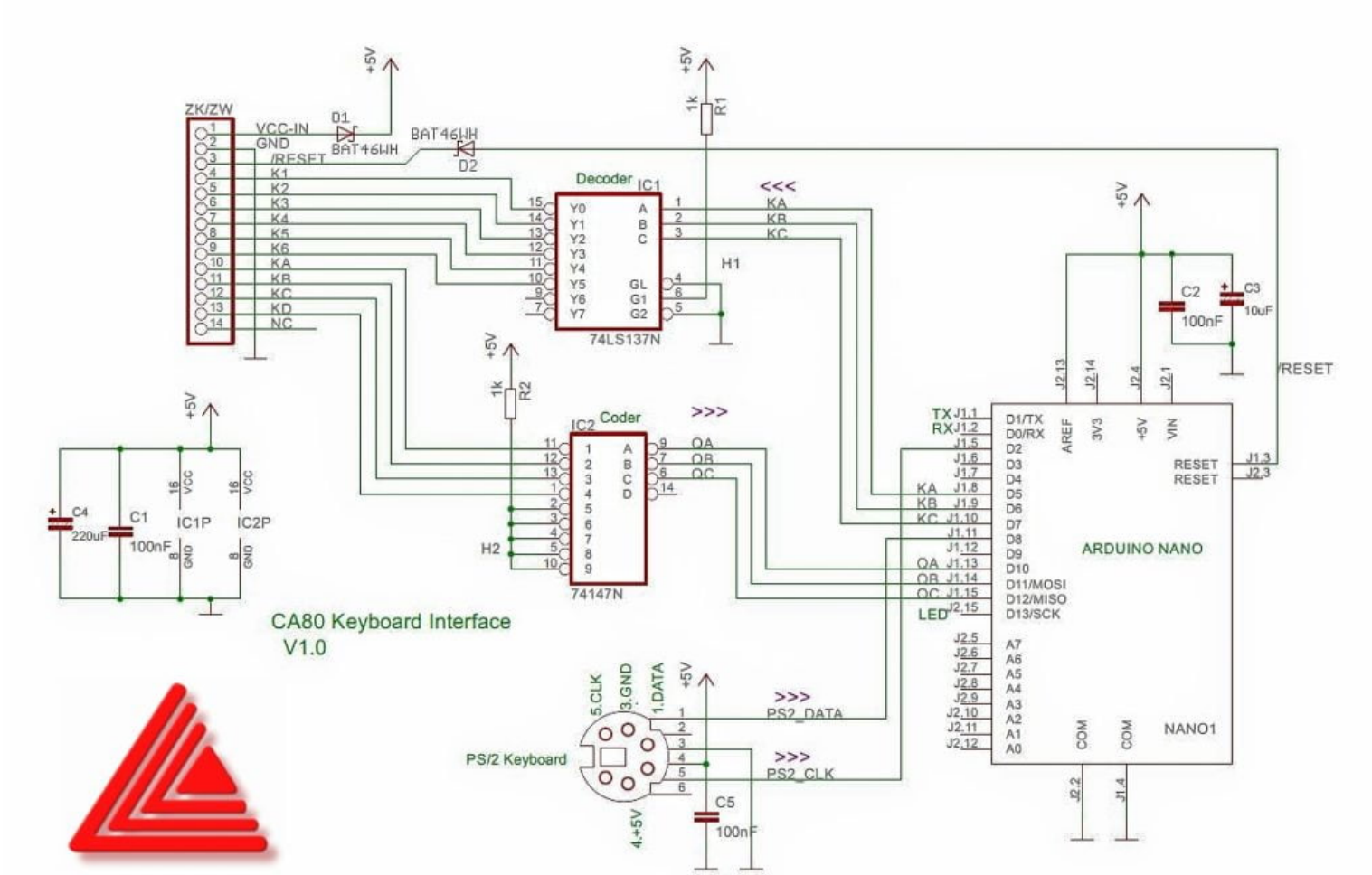
Korzystając z zimnej i deszczowej w tym roku majówki, chciałbym zaprezentować projekt interfejsu mojego autorstwa. Od niedawna stałem się bowiem szczęśliwym posiadaczem i użytkownikiem mikrokomputera retro o nazwie „CA80”, zaprojektowanego przez polskiego inżyniera Stanisława Gardynika i wylansowanego w latach 80tych ubiegłego wieku przez jego firmę MIK.

Więcej o moich perypetiach z CA80 można przeczytać TUTAJ.

Założenia projektu

1. Interfejs ma na celu zastąpienie oryginalnej klawiatury matrycowej standardową klawiaturą dla komputerów PC posiadającą interfejs PS/2 i złącze mini-DIN.
2. Zarówno klawiatura PC, jak i mikrokomputer CA80 pozostają oryginalne, brak w nich w zasadzie jakichkolwiek modyfikacji (z wyjątkiem wyprowadzenia zasilania oraz sygnału resetu na złącze klawiatury [ZK] mikrokomputera CA80 – patrz schemat ideowy interfejsu).
3. Interfejs wykonany jest na bazie ARDUINO w wersji NANO w formie dosyć kompaktowej płytki i zawiera ogólnodostępne części elektroniczne.
4. Wszystkie sygnały interfejsu są pobierane ze złącza klawiatury (ZK) mikrokomputera CA80.
5. Oprogramowanie ARDUINO jest uproszczone w maksymalnym stopniu a ilość zajętych obsługą interfejsu linii danych ARDUINO ograniczona do niezbędnego minimum.

Schemat ideowy i opis działania



Interfejs – schemat ideowy

LINK do schematu w formacie PDF.

Idea działania interfejsu może być opisana w następujących krokach prostego algorytmu:

1. Oczekiwanie mikrokontrolera Arduino na przyciśnięcie klawisza na klawiaturze PC.
2. Pobranie danych (kodu naciśniętego klawisza) z interfejsu PS/2 klawiatury.
3. Zdekodowanie kodu klawiatury PC.
4. Odczekanie aż na wyjściu przeczesywania matrycy przycisków CA80 pojawi się właściwy sygnał. Wyjścia KA, KB, KC, KD złącza klawiatury CA80 (ZK) są, poprzez koder sprzętowy IC2, sprawdzane w pętli programu Arduino i na bieżąco porównywane z odczytanym i zdekodowanym kodem uprzednio pobranym z klawiatury PC.
5. Pojawienie się poprawnego sygnału j.w. powoduje wygenerowanie przez interfejs odpowiedzi logicznej, która po zdekodowaniu sprzętowym (IC1) przekazywana jest do CA80 symulując zwarcie styków przycisku symulowanej klawiatury matrycowej.

Analizując schemat, możemy wyróżnić na nim 3 współpracujące ze sobą bloki funkcjonalne: koder (IC2), dekoder (IC1), sterownik (Arduino Nano).

- Podstawowym zadaniem kodera opartego o układ cyfrowy 74147 jest zamiana sygnałów przeczesywania matrycy CA80 z postaci 1z4 na postać liczby z przedziału 0x03 do 0x07 (patrz tablica prawdy kodera 74147). Takie rozwiązanie prócz zaoszczędzenia jednego wejścia Arduino, daje również możliwość elektrycznego buforowania sygnałów z interfejsu klawiatury CA80 zmniejszając jego obciążenie.

Value	IN1	IN2	IN3	IN4	QA	QB	QC	OUT	/OUT
0	1	1	1	1	1	1	1	0x07	0x00
1	0	1	1	1	0	1	1	0x06	0x01
2	1	0	1	1	1	0	1	0x05	0x02
3	1	1	0	1	0	0	1	0x04	0x03
4	1	1	1	0	1	1	0	0x03	0x04

Tablica prawdy kodera 74147

Zgodnie z tablicą prawdy układu kodera 74147 zarówno sygnały na wejściu jak wyjściu tego układu są zanegowane, czyli stanem aktywnym jest poziom niski „L”.

Jest możliwe pominięcie układu kodera 74147, w takim przypadku sygnały wyjściowe przeczesywania klawiatury (KA do KD) należy podłączyć bezpośrednio na wejścia modułu kontrolera Arduino.
W takim przypadku konieczna jest m.in. zmiana tablicy kodowania zawartej w programie interfejsu (funkcja getKeyString()).

- Dekoder IC1 oparty na układzie 74137 zamienia sygnały sterujące z modułu Arduino na postać 1z6. Rozwiązanie takie upraszcza sterowanie i redukuje ilość zajętych wyjść Arduino z 6 do 3.

value	/value	A	B	C	Y0	Y1	Y2	Y3	Y4	Y5
0x07	0x00	1	1	1	1	1	1	1	1	1
0x00	0x07	0	0	0	0	1	1	1	1	1
0x01	0x06	1	0	0	1	0	1	1	1	1
0x02	0x05	0	1	0	1	1	0	1	1	1
0x03	0x04	1	1	0	1	1	1	0	1	1
0x04	0x03	0	0	1	1	1	1	1	0	1
0x05	0x02	1	0	1	1	1	1	1	1	0

Tablica prawdy dekodera 74137

Tablica prawdy kodera 74137 pokazuje, że jego sygnały wejściowe nie są zanegowane (aktywny jest tutaj stan wysoki „H”), w przeciwieństwie do sygnałów wyjściowych, gdzie aktywny jest niski stan sygnału – „L”.

Program dla Arduino

Sketch dla Arduino w spakowanym pliku *.INO jest do pobrania **TUTAJ**.

```
//-------------------------------------------------------------------------------
-----------------------------------------
//      CA80 PS2 Keyboard Transcoder v2.3P (wersja produkcyjna z polskimi
komentami
//       PCB v1.0 (Arduino NANO)
//-------------------------------------------------------------------------------
-----------------------------------------
//  Autor: Sławomir Jurkiewicz (elserw@elserw.com)
//  Na podstawie: "Podłączamy stara klawiaturę do Arduino" by Kamil
(https://starter-kit.nettigo.pl/author/kamil/feed/)
//    oraz "Zdalne sterowanie klawiaturą" by Natasza Biecek
(http://bienata.waw.pl/ca808.php)
//-------------------------------------------------------------------------------
```

```
------------------------------------------

//Definicja funkcji pinów Arduino
#define CLK 2 //inp
#define DAT 8 //inp

#define QA 10 //inp
#define QB 11 //inp
#define QC 12 //inp

#define LED 13 //out
#define KA 5  //out
#define KB 6  //out
#define KC 7  //out

//Deklaracja i zerowanie bufora obsługi danych z klawiatury PC PS/2
const int BUF_SIZE = 11;
bool buffer[BUF_SIZE] = {0};
//Pozostałe zmienne dla procedur PS/2
int pos = 0;
bool ignoreNext = false;
unsigned long lastRead = 0;

int caCode = 0; //Bieżący kod naciśniętego klawisza wysyłany do CA80

void setup() {
  //Komunikacja monitora portu szeregowego Arduino do celów debugowania
programu
  //Wszystkie linijki kodu zaczynające się od "Serial." można usunąć w gotowym
programie
  Serial.begin(9600);
  Serial.println("CA80 PS2 Keyboard Transcoder v2.3P");
  Serial.println("Ready...");

  //Ustawienie funkcji zdefiniowanych pinów w Arduino
  //Wejścia
  pinMode(CLK, INPUT);
  pinMode(DAT, INPUT);
  pinMode(QA, INPUT);
  pinMode(QB, INPUT);
  pinMode(QC, INPUT);
  //Wyjścia
  pinMode(LED, OUTPUT);
  pinMode(KA, OUTPUT);
  pinMode(KB, OUTPUT);
  pinMode(KC, OUTPUT);

  digitalWrite(LED, LOW); //Wygaszenie LEDa na płytce Arduino.
  PORTD = 0xE0;  //Ustawienie "1" na pinach D5,D6,D7 (KA, KB, KC). Dekoder IC1
ma wszystkie wyjścia Y0 do Y7 ustawione na "1"
```

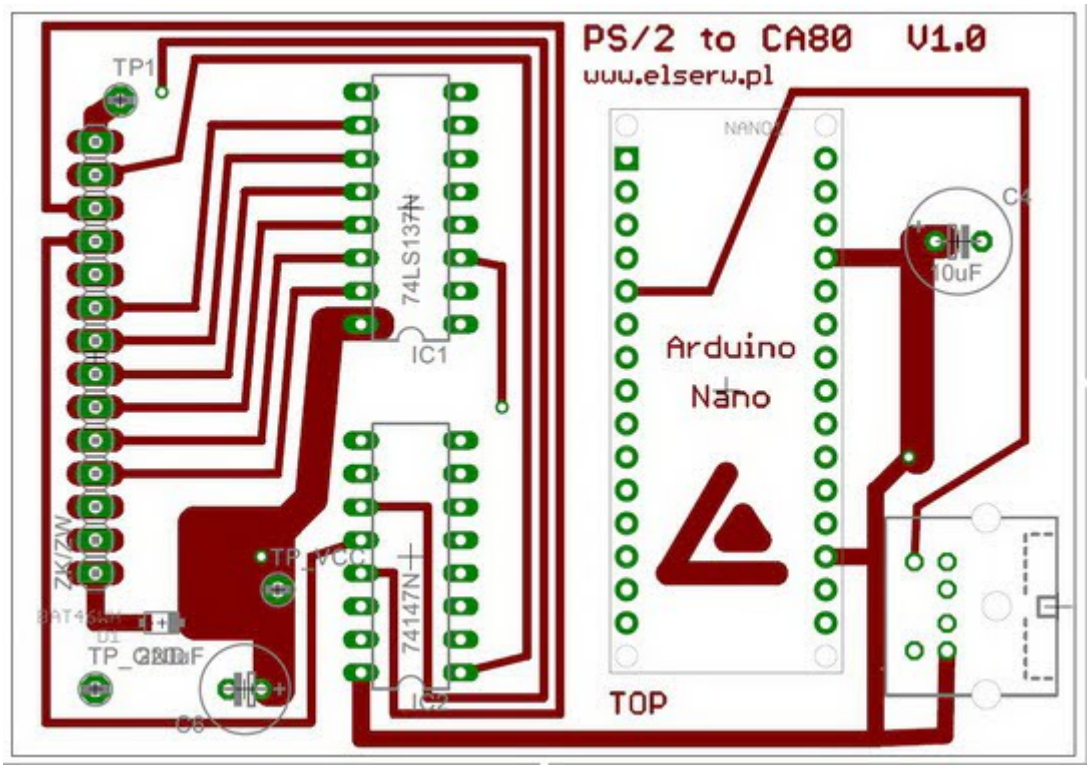
```
  delay(1000); //Oczekiwanie na zadziałanie klawiatury PS/2 po włączeniu zasilania
  attachInterrupt(digitalPinToInterrupt(CLK), readData, FALLING); //Ustawienie przerwania od wyjścia zegarowego CLK klawiatury PS/2 (reakcja na zbocze dodatnie sygnału CLK)
}

//Główna pętla programu - wczytanie, dekodowanie klawisza z PS/2 oraz zakodowanie go i wysłanie do CA80 (poprzez ustawienie w odpowiednim momencie wyjść dekodera IC1)
void loop() {
  if(pos != 0 && millis() - lastRead > 1000) {  //oczekiwanie na zapełnienie bufora klawiatury, którego zawartość jest kompletowana w przerwaniu (procedura "readData").
    pos = 0;
  }
  if(pos == 11) { //bufor PS/2 kompletny, wczytano kod klawisza z klawiatury PS/2
    pos = 0;
    int keyCode = getKeyCode(buffer); //Obróbka danych z bufora. W zmiennej "keyCode" jest kod klawisza z PS/2
    if(ignoreNext) {
      ignoreNext = false;
      return;
    }
    if(keyCode == 0xF0) { //Ignoruj kody puszczenia klawisza na PS/2
      ignoreNext = true;
      return;
    }
    //Wysłanie na port szeregowy terminala Arduino informacji o kodzie i nazwie wciśniętego przycisku
    Serial.print("PS/2=0x");
    Serial.print (String(keyCode, HEX));
    Serial.print (" ");
    String keyString=getKeyString(keyCode); //Zdekodowanie kodu klawisza PS/2 na jego nazwę (przy okazji w zmiennej "caCode" znajduje się kod klawisza dla CA80)
    Serial.print(keyString);

    if(caCode>0) {
      //Wysłanie na terminal Arduino informacji o kodzie dla CA80 wciśniętego klawisza na klawiaturze PS/2
      Serial.print (" CA80=0x");
      Serial.print (String(caCode, HEX));
      noInterrupts(); //zatrzymanie przwań
      sendKey(caCode); //wysłanie do CA80 kodu naciśniętego klawisza
      interrupts(); //wznowienie obsługi przerwań
      digitalWrite(LED, !digitalRead(LED)); //zmiana świecenia LED - sygnalizuje wczytanie z klawiatury i wysłanie do CA jednego kodu
      Serial.print (" ");
      pos = 0; //po włączeniu przerwań bufor klawiatury PS/2 będzie czytany od
```

```
nowa
    }
    Serial.println ("");
  }
}

void sendKey(int caCode) {
  //wysłanie kodu naciśniętego klawisza do CA80 (kod  znajduje się w zmiennej
"caCode")
  //https://www.arduino.cc/en/Reference/PortManipulation

  int caCodeT = (caCode & 0xF0) >> 4; //modyfikacja zdekodowanego kodu
klawisza

  for (int i = 0; i <= 20; i++) { //powtórz 20 razy wysłanie klawisza do CA80,
aby przechytrzyć debouncing
    while (caCodeT!=(PINB & 0x1C) >> 2) { //czytaj porty 10,11,12 (QA, QB, QC)
aby ustalić moment kiedy należy ustawić sygnały KA,KB,KC
      //pętla oczekiwania na właściwy moment, czyli wystawienie przez CA80
odpowiednich sygnałów KA, KB, KC, KD
    }
    PORTD = (caCode & 0x07)<<5; //ustaw porty D5,D6,D7 (KA, KB, KC) w celu
wysłania kodu do CA80
    delayMicroseconds(12); //podtrzymaj ustawienia do czasu kiedy CA80 skończy
dekodowanie wysłanych mu danych - czas 12us jest krytyczny dla całego procesu
    PORTD = 0xE0;  //zapisz "111" do portów D5,D6,D7 w celu wyzerowania (KA,
KB, KC)
  }
}


void readData() {
  //Sczytanie danych z klawiatury PS/2
  lastRead = millis();
  buffer[pos++ % 11] = digitalRead(DAT);
}

int getKeyCode(bool * buf) {
  //Na podstawie danych z bufora funkcja zwraca kod naciśniętego klawisza.
  bool parity = 1;
  int result = 0;
  if(buf[0] != 0) return -1;
  if(buf[10] != 1) return -2;
  for(int x = 0; x < 8; x++) {
    result |= buf[1+x] << x;
    if(buf[1+x]) parity = !parity;
  }
  if(buf[9] != parity) return -3;
  return result;
}
```

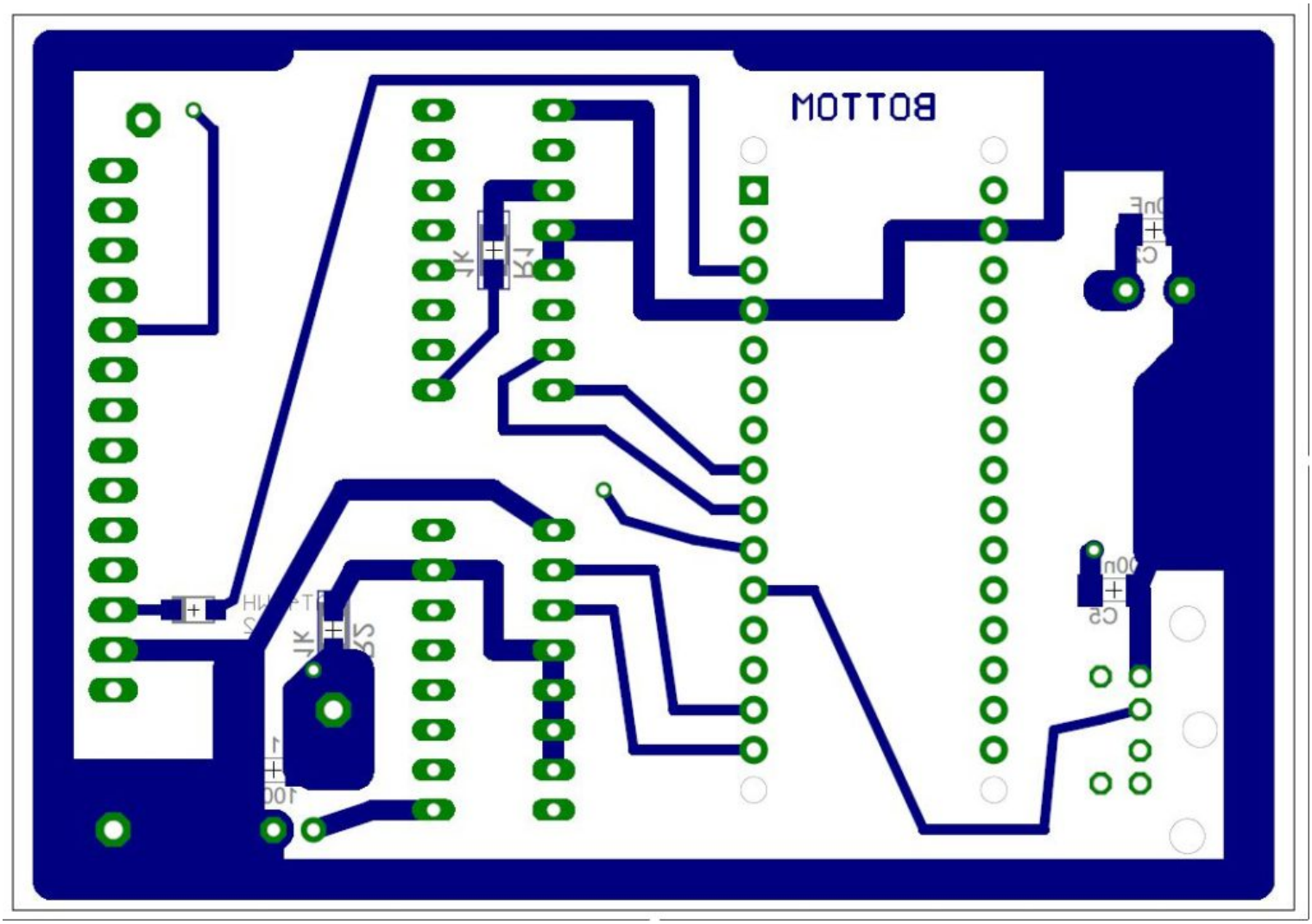
```
String getKeyString(int keyCode) {
  //Dekodowanie naciśniętego klawisza na podstawie jego kodu
  //  przy okazji w zmiennej "caCode" jest ustawiany kod dla CA80
  caCode=0;
  switch(keyCode) {
    case 0x1C: caCode=0x52; return "A";
    case 0x32: caCode=0x51; return "B";
    case 0x21: caCode=0x64; return "C";
    case 0x23: caCode=0x63; return "D";
    case 0x24: caCode=0x62; return "E";
    case 0x2B: caCode=0x61; return "F";
    case 0x34: caCode=0x50; return "G";
    //case 0x33: return "H";
    //case 0x43: return "I";
    //case 0x3B: return "J";
    //case 0x42: return "K";
    //case 0x4B: return "L";
    case 0x3A: caCode=0x60; return "M";  //
    //case 0x31: return "N";
    //case 0x44: return "O";
    //case 0x4D: return "P";
    //case 0x15: return "Q";
    //case 0x2D: return "R";
    //case 0x1B: return "S";
    //case 0x2C: return "T";
    //case 0x3C: return "U";
    //case 0x2A: return "V";
    //case 0x1D: return "W";
    //case 0x22: return "X";
    //case 0x35: return "Y";
    //case 0x1A: return "Z";
    case 0x45: caCode=0x34; return "0";
    case 0x16: caCode=0x33; return "1";
    case 0x1E: caCode=0x32; return "2";
    case 0x26: caCode=0x31; return "3";
    case 0x25: caCode=0x44; return "4";
    case 0x2E: caCode=0x43; return "5";
    case 0x36: caCode=0x42; return "6";
    case 0x3D: caCode=0x41; return "7";
    case 0x3E: caCode=0x54; return "8";
    case 0x46: caCode=0x53; return "9";
    case 0x5A: caCode=0x30; return "ENT"; //"=" <ENTER>
    case 0x55: caCode=0x30; return "="; //"=" <ENTER>
    case 0x29: caCode=0x40; return "SPC"; //"." <SPACE>
    case 0x49: caCode=0x40; return "."; //"."
    case 0x05: caCode=0x65; return "F1";  //
    case 0x06: caCode=0x55; return "F2";  //
    case 0x04: caCode=0x45; return "F3";  //
    case 0x0c: caCode=0x35; return "F4";  //
    case 0x76: caCode=0x60; return "ESC";  //"M" <ESC>
  }
```

```
  return "";
}
```

Płytka drukowana

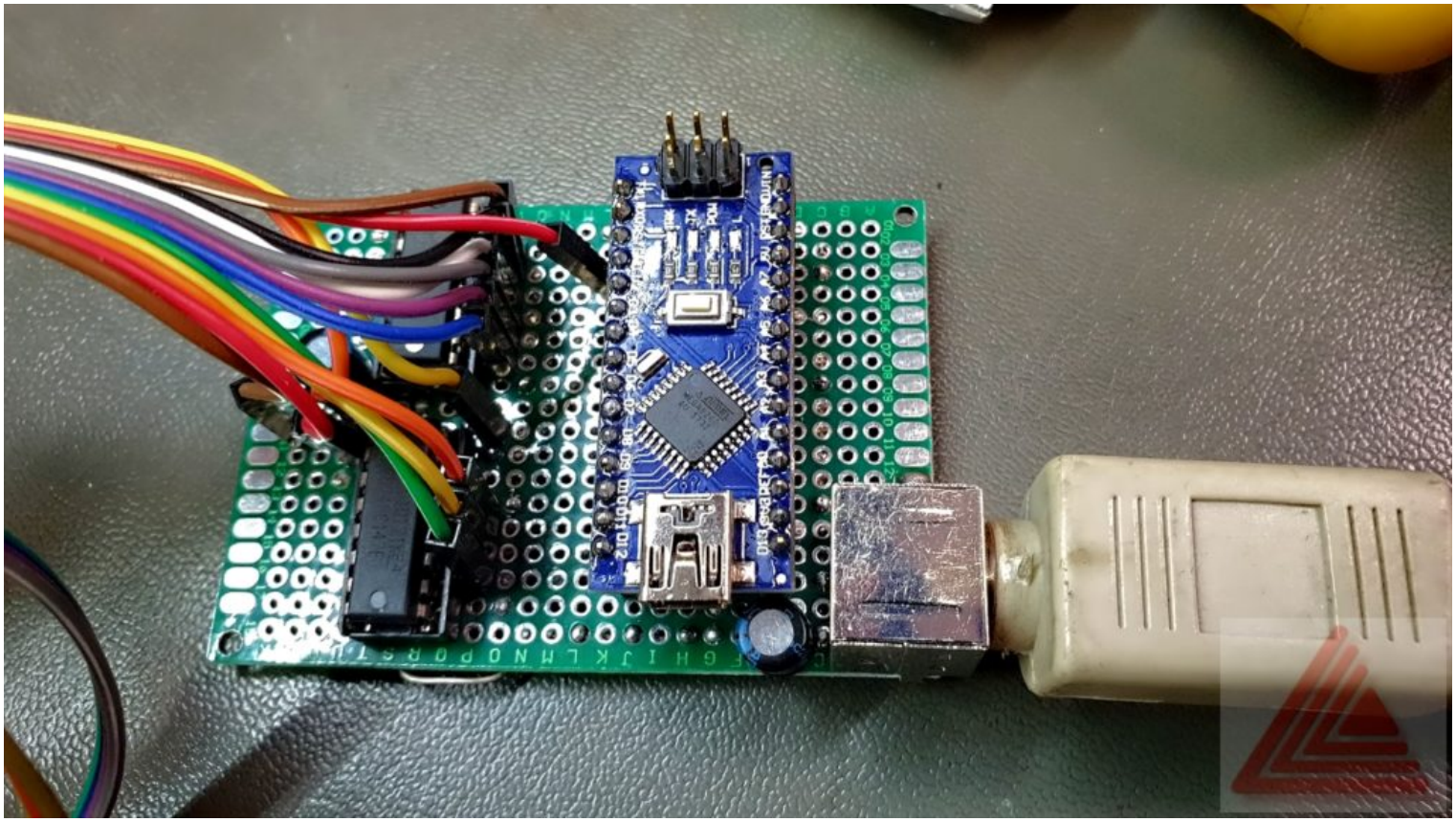


Strona elementów (TOP)

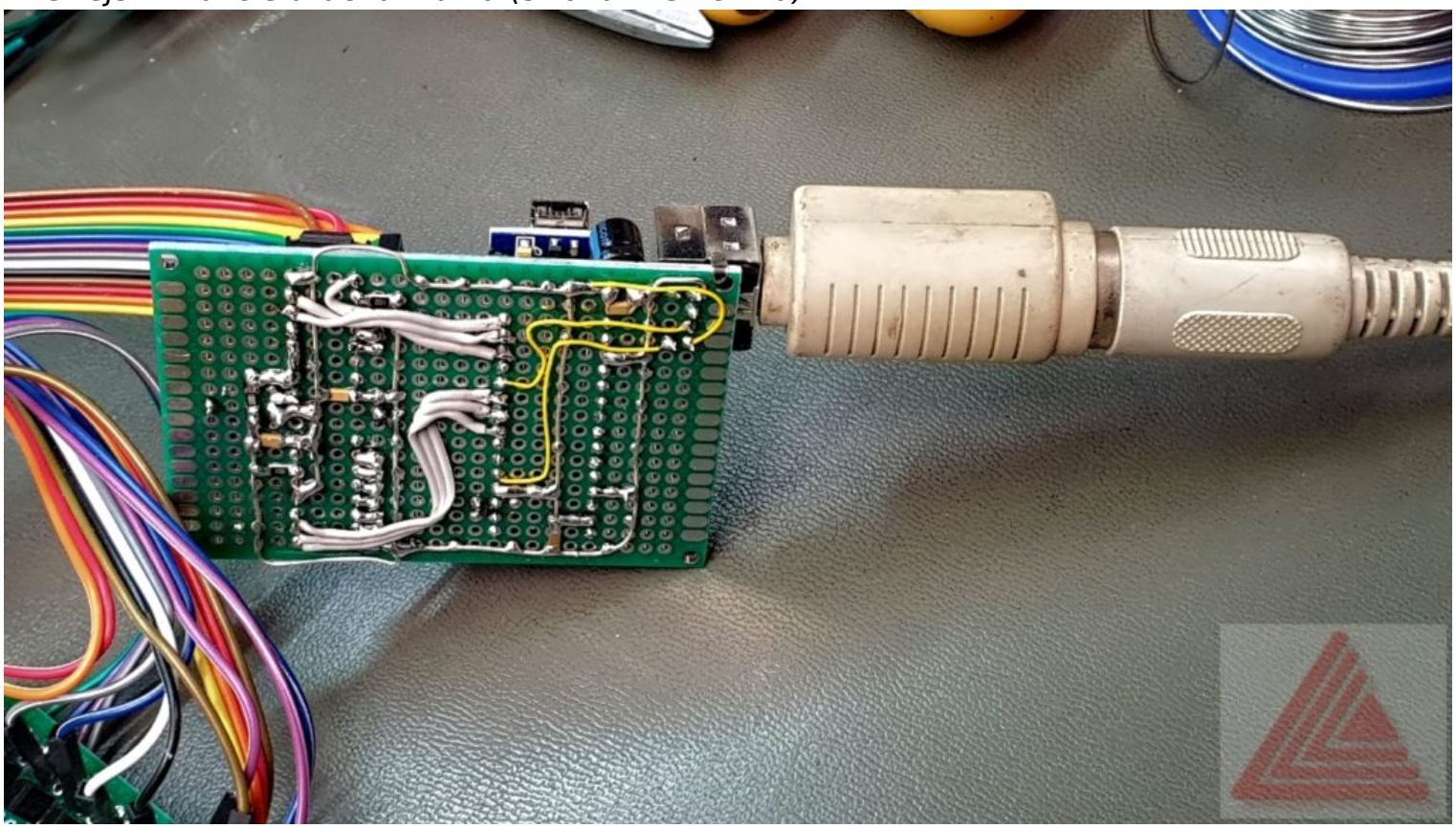


Strona połączeń (BOTTOM)

Interfejs w trakcie uruchamiania.



Interfejs w trakcie uruchamiania (strona wierzchnia).

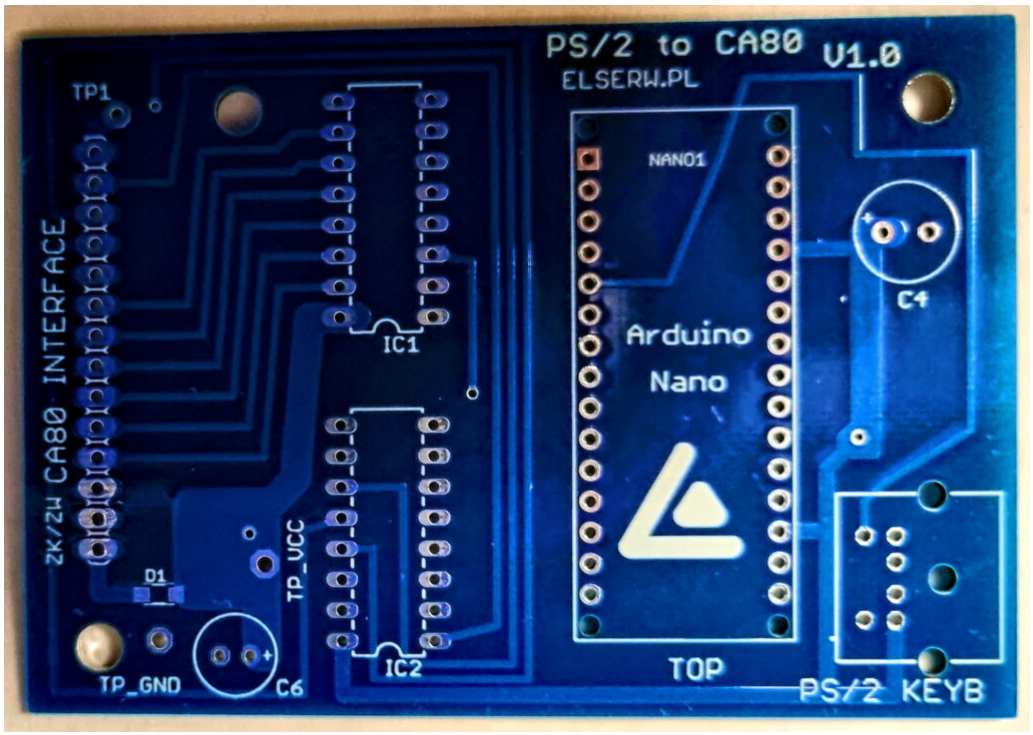


Interfejs w trakcie uruchamiania (strona spodnia).

W czasie tworzenia projektu wykorzystano następujące źródła:

1. *„Podłączamy starą klawiaturę do Arduino”* by Kamil
2. *„Zdalne sterowanie klawiaturą”* by Natasza Biecek
3. *„The PS/2 Mouse/Keyboard Protocol”* (Link)

P.S. Posiadamy kilka profesjonalnych płytek PCB dla tego projektu. Zainteresowanych prosimy o kontakt mailowy lub telefoniczny.



PCB interfejsu klawiatury PS2 dla CA80