

Karta projektu zaliczeniowego części zadaniowej egzaminu

Systemy mikroprocesorowe - 2020

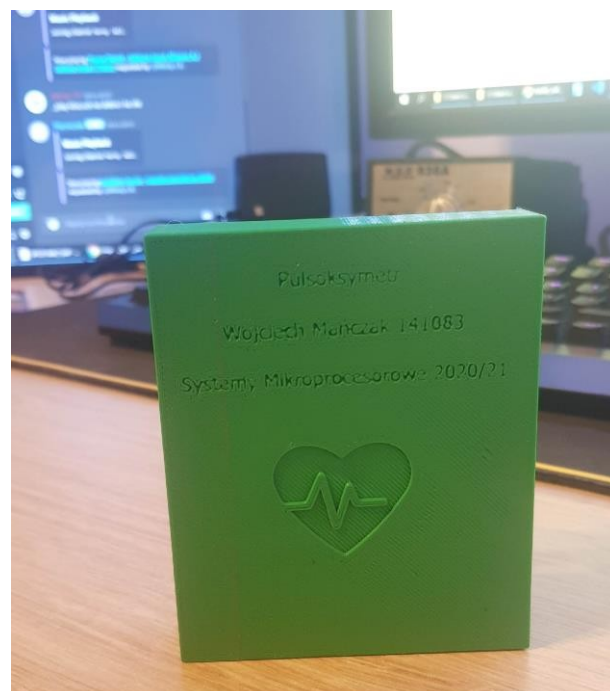
Temat projektu: **Pulsoksymetr**

Imię i nazwisko:

Nr albumu:

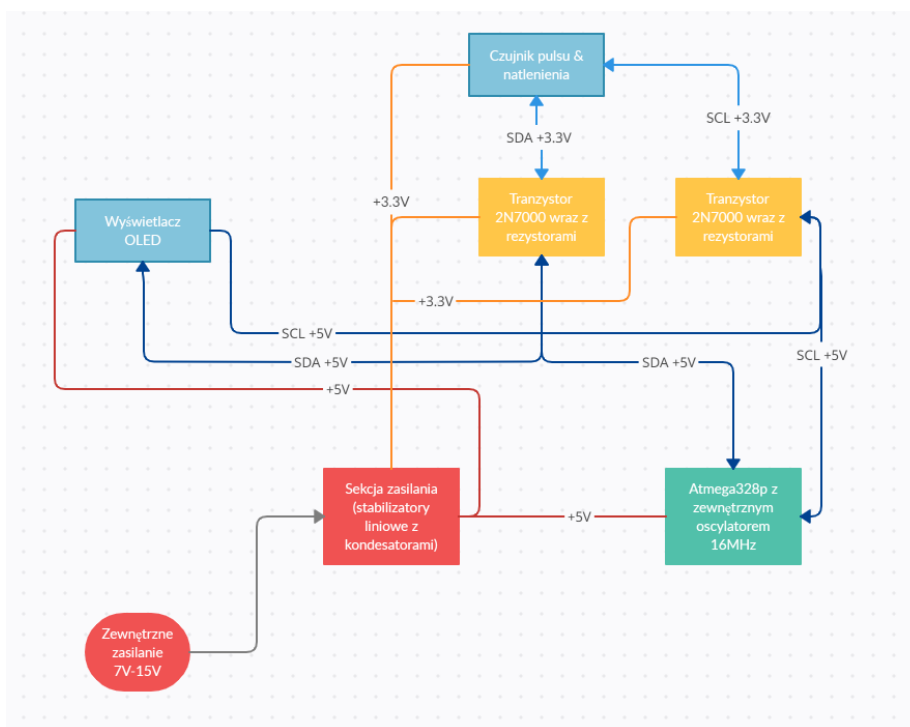
Kierunek: **AiR**, grupa: **A3**,

1. Opis projektu



Pomysł na projekt powstał w czasie izolacji domowej spowodowanej pozytywnym wynikiem SARS-CoV-2. Jednym z zaleceń WHO jest systematyczne mierzenie natlenienia krwi przy użyciu pulsoksymetrów. Z powodu pandemii, ich ilość została ograniczona, dlatego postanowiłem zbudować go własnoręcznie. Układ opiera się o czujnik pulsu i natlenienia krwi - MAX30100 oraz wyświetlacz OLED z kontrolerem I2C SH1106. Całość kontrolowana przez Atmega328p z bootloader'em arduino. Zasilanie z zewnętrznego źródła poprzez złącze DC Jack 5.5 mm, stabilizowane przez układ do potrzebnych napięć. Całość zamknięta w obudowie wydrukowanej na drukarce Ender 3 PRO oraz zaprojektowanej w Autodesk Inventor.

Rysunek 1. 1 Schemat blokowy systemu



2. Hardware

Tabela 2. 1 Spis najważniejszych elementów

	Atmega 328p
	Pulsometr MAX30100 wraz z regulatorem 3.3V -> 1.8V napięcia oraz peryferiami. Komunikacja przy użyciu I2C.
	Wyświetlacz OLED, wraz z wbudowanym kontrolerem SH1106. Komunikacja przy użyciu I2C.
	Stabilizator liniowy 5V L7805CV
	Stabilizator liniowy LDO 3,3V LD1117V33
	Tranzystor N-MOSFET 2N7000

Dodatkowo, aby zagwarantować poprawną pracę powyższych elementów, użyto:

Złącze DC Jack 5.5mm/2.1mm

Kondensatory ceramiczne 100nF

Kondensatory ceramiczne 22pF

Podstawka do układów DIP 28-pinowa

LED 5mm czerwony

Oscylator kwarcowy 16MHz

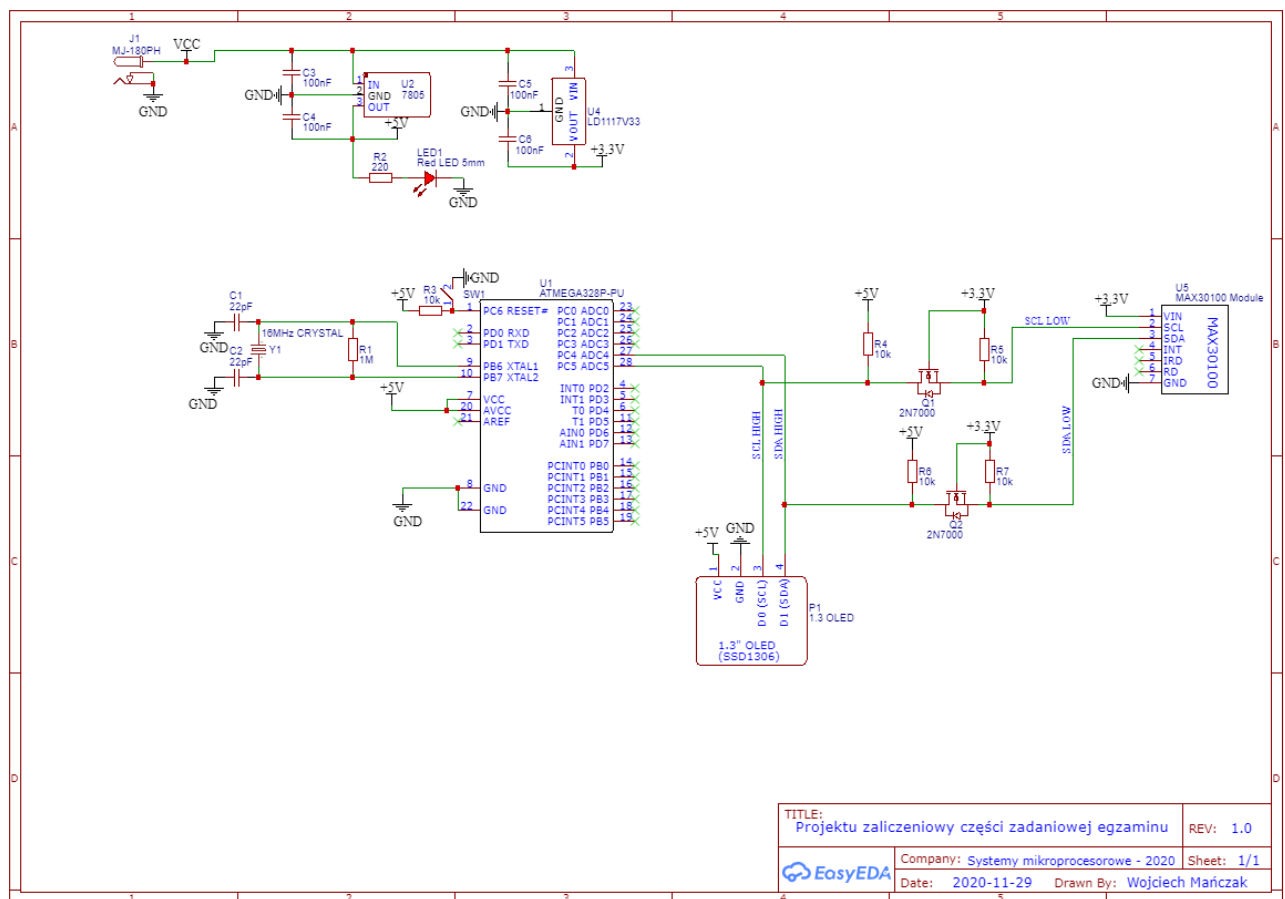
Tact switch - 2 pinowy

Rezystor 10kΩ

Rezystor 220Ω

Schemat elektroniczny układu, zaprojektowany oraz narysowany przy pomocy narzędzia EasyEDA¹:

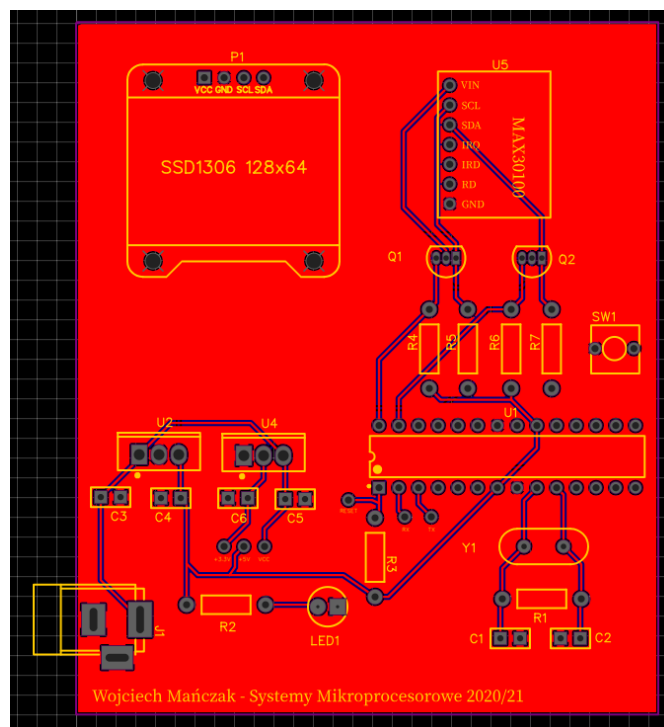
Rysunek 2. 1 Schemat elektroniczny układu



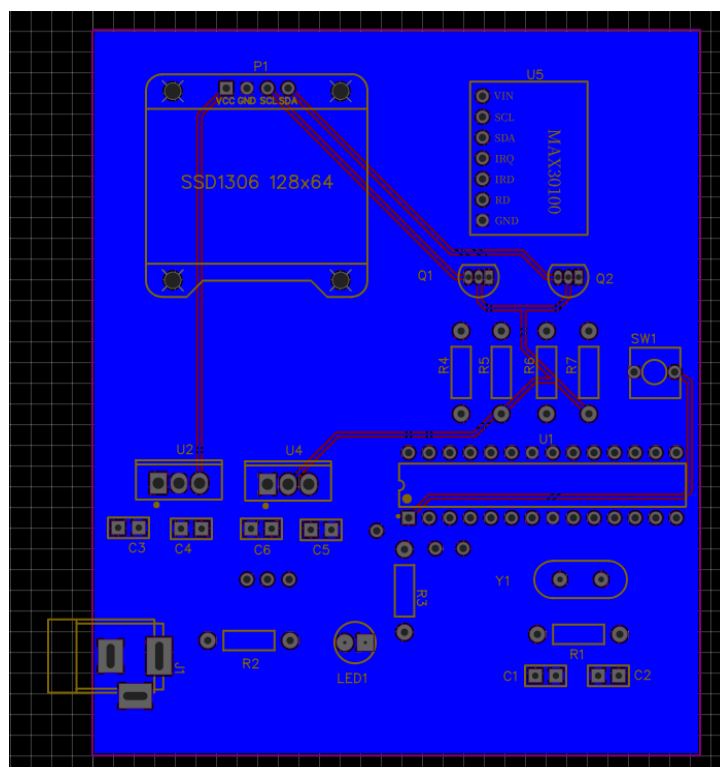
¹ EasyEDA – Symulator obwodów i konstrukcja PCB Online - <https://easyeda.com>

Schemat płytki drukowanej, zaprojektowany przy pomocy narzędzia EasyEDA oraz wytworzonej przez firmę JLCPCB²:

Rysunek 2. 2 Górna warstwa płytki drukowanej

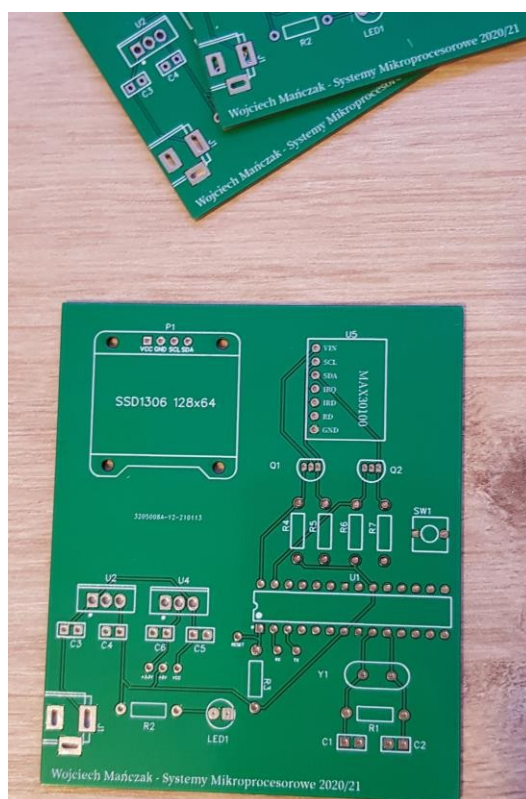


Rysunek 2. 3 Dolna warstwa płytki drukowanej

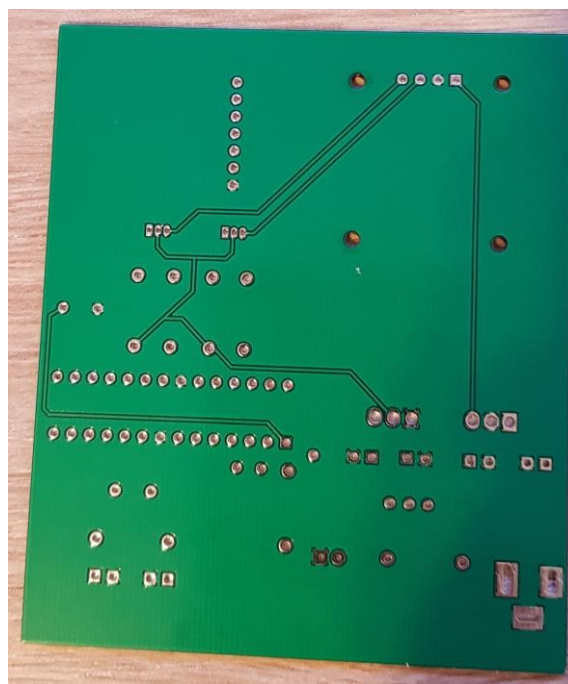


² JLCPCB PCB Manufacturing - <https://jlcpcb.com>

Rysunek 2. 4 Front płytki drukowanej

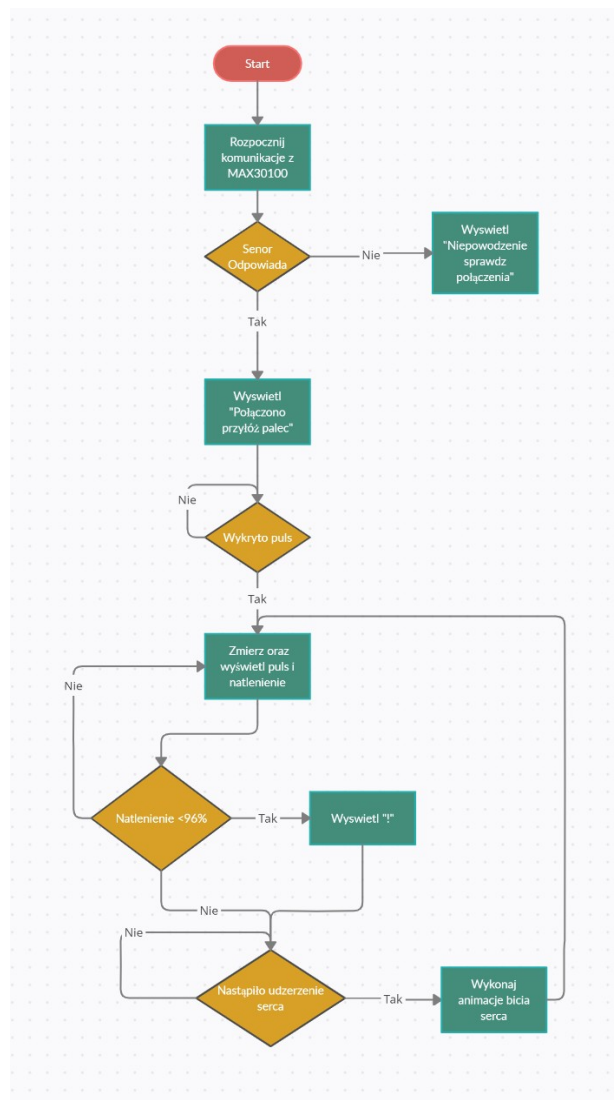


Rysunek 2. 5 tył płytki drukowanej



3. Software

Rysunek 3. 1 Schemat Blokowy programu



Ponieważ atmega328p użyta w układzie została wypalona z bootloaderem arduino korzystamy ze środowiska Arduino IDE oraz edytora Visual Studio Code. Dzięki temu mamy dostęp do bibliotek obsługujących czujnik pulsu oraz wyświetlacz OLED.

Wykorzystane biblioteki:

Wire.³ - komunikacja po I2C

MAX30100_PulseOximeter.h⁴ - obsługa pomiarów pulsu oraz natlenienia MAX30100

MAX30100.h⁵ - obsługa wbudowanych zabezpieczeń czujnika MAX30100

U8g2lib.h⁶ - biblioteka do graficznej obsługi monochromatycznych wyświetlaczy

³ Biblioteka Arduino - <https://www.arduino.cc/en/reference/wire>

⁴ Biblioteka dostarczona przez producenta - OXullo Intersecans - https://github.com/oxullo/Arduino-MAX30100/blob/master/src/MAX30100_PulseOximeter.h

⁵ Biblioteka dostarczona przez producenta - OXullo Intersecans - <https://github.com/oxullo/Arduino-MAX30100>

⁶ Biblioteka stworzona przez użytkownika – olikraus - <https://github.com/olikraus/u8glib>

Podczas budowania prototypu układu na płytce stykowej, posługiwałem się Arduino Uno, które po skończeniu prac, posłużyło również do wypalenia bootloadera oraz wgrania programu do Atmegi 328p. Całość wykonana została przy użyciu środowiska Arduino IDE oraz wbudowanego w Arduino UNO programatora.

Najważniejsze elementy programu:

Rysunek 3. 2 Rozpoczęcie komunikacji oraz ustawienie parametrów sugerowanych przez producenta

```
void setup()
{
    u8g2.begin();
    init_display();

    pox.begin();
    pox.setOnBeatDetectedCallback(onBeatDetected);
    pox.setIRLedCurrent(LED_CURRENT);
    sensor.setMode(MAX30100_MODE_SPO2_HR);
    sensor.setLedsPulseWidth(PULSE_WIDTH);
    sensor.setSamplingRate(SAMPLING_RATE);
}
```

Rysunek 3. 3 Obsługa połączenia z czujnikiem MAX30100

```
if (!pox.begin()) {
    u8g2.setCursor(40,12);
    u8g2.print("Niepowodzenie");
    u8g2.setCursor(15,29);
    u8g2.print("Sprawdz czujnik!");
    u8g2.sendBuffer();
    for(;;);
} else {
    u8g2.setCursor(20,12);
    u8g2.print("Polaczona");
    u8g2.setCursor(0,29);
    u8g2.print("Przyloz palec...");
    u8g2.sendBuffer();
}
```

Rysunek 3. 4 Pomiar pulsu oraz natlenienia

```
pox.update();
HRclean = pox.getHeartRate();
SpO2 = pox.getSpO2();
```

Rysunek 3. 5 wyświetlanie pulsu oraz natlenienia

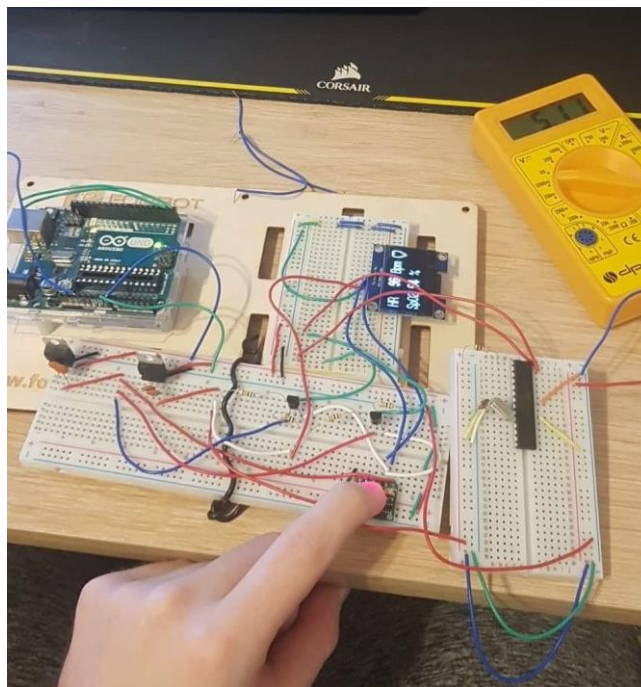
```
u8g2.setFont(u8g2_font_fub11_tf);
u8g2.setCursor(45,12);
u8g2.print(HRclean);
u8g2.setCursor(45,30);
u8g2.print(SpO2);
```


4. Proces montażu i programowania systemu

Proces budowy rozpoczął się od narysowania schematu ideowego, zapoznania się z dokumentacją a następnie narysowaniu schematu elektrycznego.

Następnie powstał prototyp układu. Początkowo oparty o logikę Arduino UNO, lecz w późniejszych krokach pojawiła się już Atmega328p z bootlooaderem Arduino (Uno służy tutaj jedynie jako połączenie do zasilacza 12V).

Rysunek 5. 1 Pierwszy działający prototyp

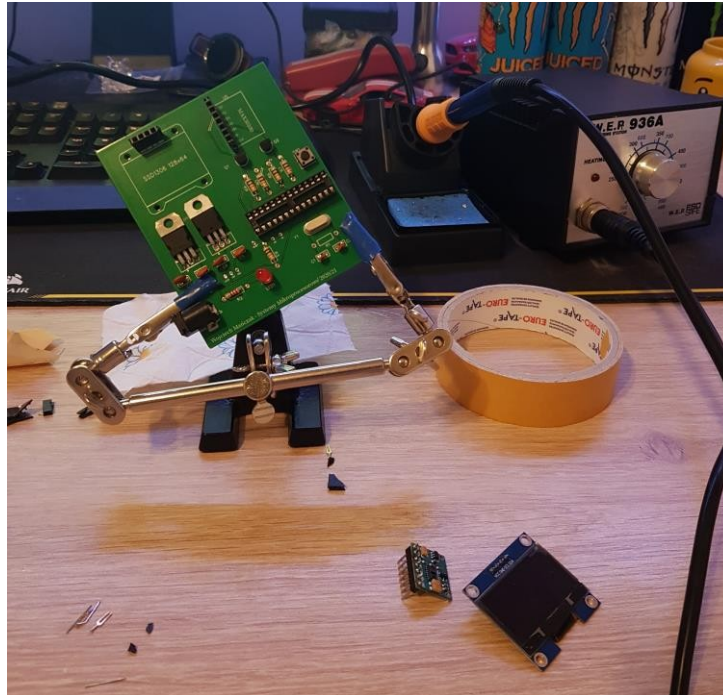


Następnie zaprojektowana i zamówiona została płytki PCB:

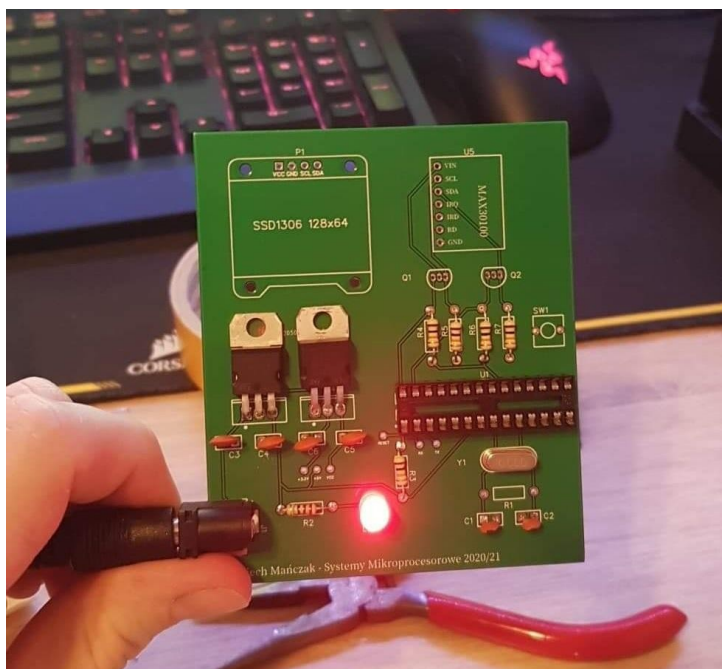
Rysunek 5. 2 Odbiór płytek PCB



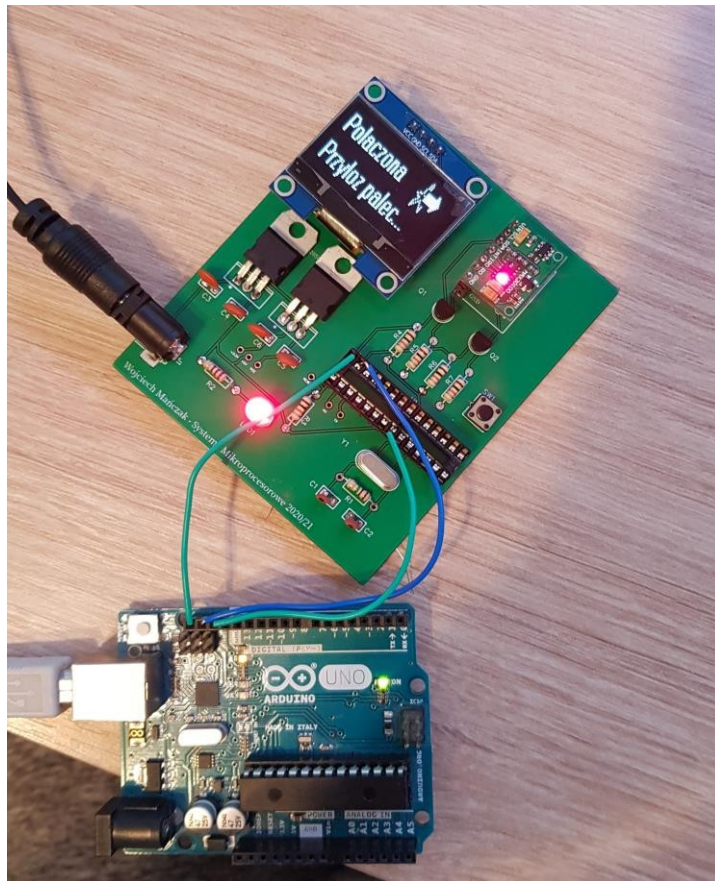
Rysunek 5. 3 Proces przylutowywania elementów do PCB



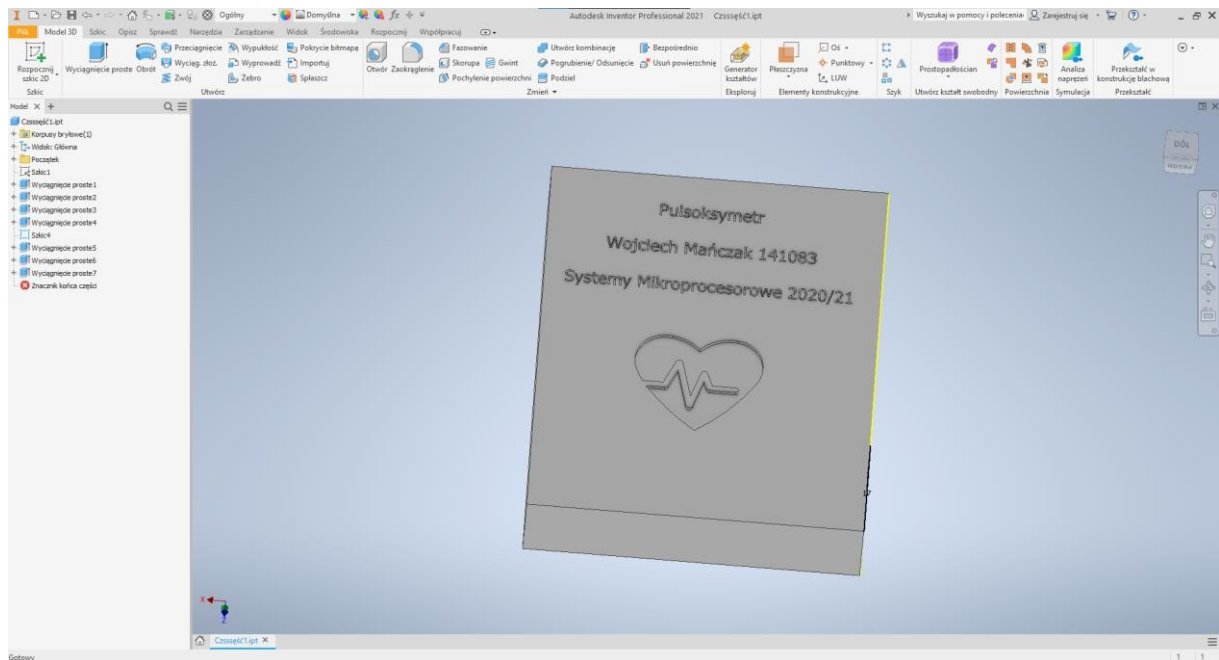
Rysunek 5. 4 Pierwsze testy sekcji zasilania



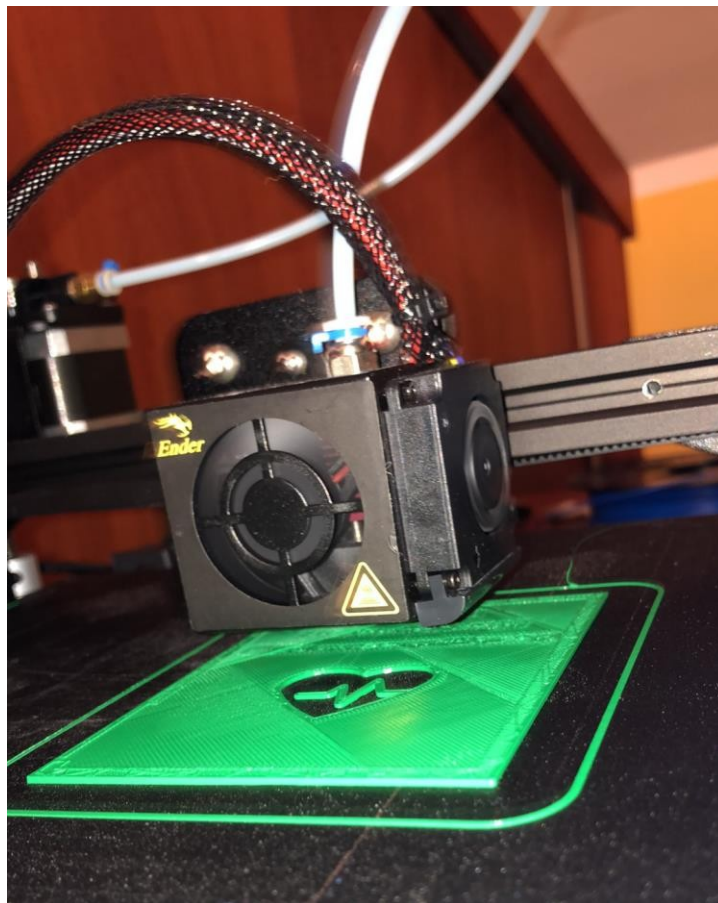
Rysunek 5. 5 Ostateczne poprawki w kodzie



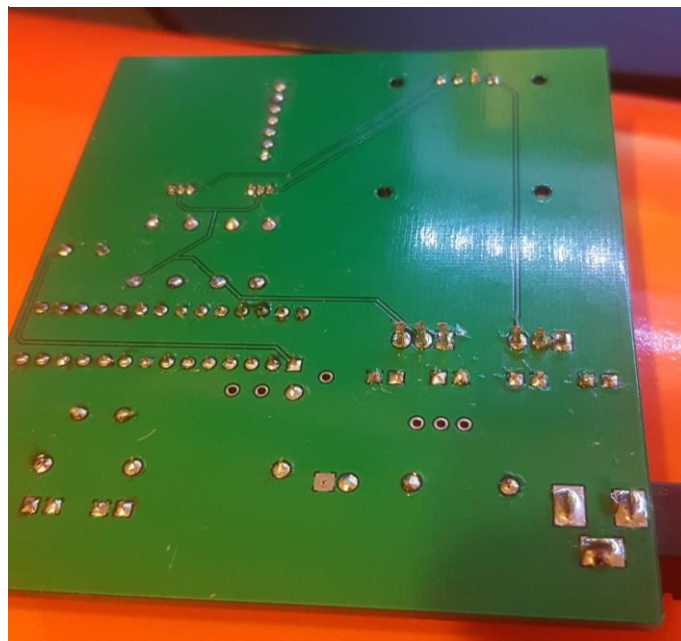
Rysunek 5. 6 Projektowanie obudowy w Inventorze



Rysunek 5. 7 Drukowanie obudowy

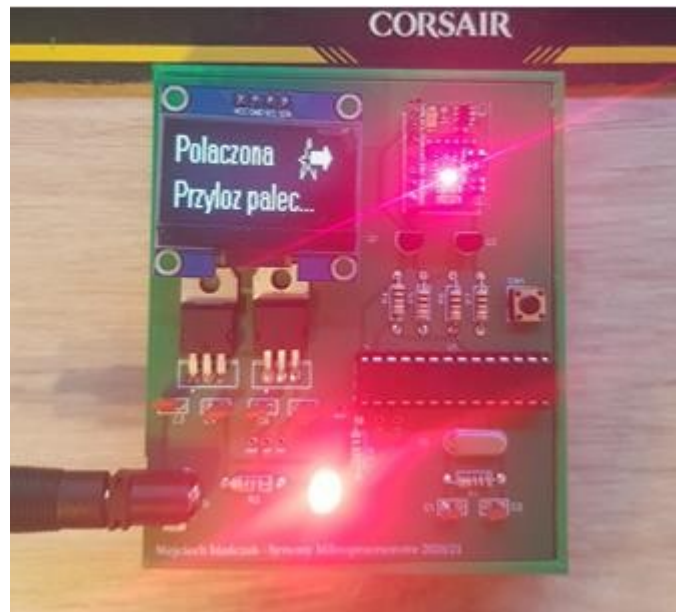


Rysunek 5. 7 Ostateczna inspekcja połączeń

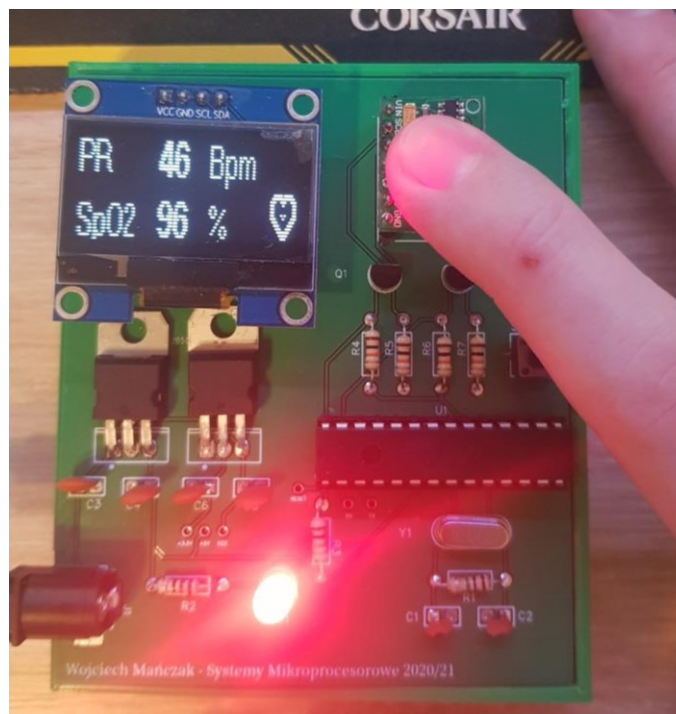


5. Weryfikacja poprawności działania układu

Rysunek 5. 8 Poprawne połączenie z czujnikiem, oczekiwanie na użytkownika



Rysunek 5. 9 Wykonywanie pomiaru



6. Obsługa układu

Obsługa układu rozpoczyna się od podłączenia zasilania do portu DC jack. Układ może być zasilany z zewnętrznego zasilacza (Max 15V z powodu nie chłodzonej sekcji zasilania) lub z baterii 9V.

Po dostarczeniu zasilania układ uruchamia się samodzielnie. Dokonuje próby połączenia z czujnikiem, i prosi użytkownika o przyłożenie palca. Należy wtedy przyłożyć palec do czerwonej diody LED, i nie poruszać nim. Wyświetlone zostanie serce jeżeli użytkownik prawidłowo umieścił palec. Układ wykonuje pomiary w czasie rzeczywistym, więc chwilowe zakłamanie wartości zostaną szybko skorygowane. Jeżeli układ wykryje za niskie natlenienie krwi wyświetli ostrzeżenie w postaci „!”. Dodatkowo w momencie uderzenia serca (monitorowanie na żywo) wykonana zostanie animacja pulsowania serca.

7. Literatura

Forum robotyczne Forbot - <https://forbot.pl/blog/>

Symulator obwodów i konstrukcja PCB Online - <https://easyeda.com>

Strona platformy Arduino - <https://www.arduino.cc/en>

Strona GitHub producenta MAX30100 - OXullo Intersecans - <https://github.com/oxullo>

Strona GitHub użytkownika – olikraus - <https://github.com/olikraus>