

**Dokumentacja techniczna oraz
instrukcja obsługi urządzenia.**

TIMDIS

- urządzenie do odmierzania czasu mycia zębów.

Spis treści

1. Wstęp

1.1 Opis urządzenia

2. Instrukcja obsługi urządzenia

3. Dane techniczne urządzenia

3.1 Użyte komponenty

3.2 Uproszczony rysunek techniczny

3.3 Kod programu

1. Wstęp

1.1 Opis urządzenia

Urządzenie Timdis służy do ustawiania oraz przeprowadzania pomiaru czasu mycia zębów.

Urządzenie składa się z wyprofilowanego uchwytu na pojemnik typu kubek o średnicy (60 mm) oraz panelu sterowania do wprowadzania ustawień czasu.

Panel sterowania posiada trzy przyciski, wyświetlacz oraz zewnętrzny brzęczyk do wydawania dźwięku.

2. Instrukcja obsługi

1. Ustawiamy urządzenie w dogodnym dla siebie miejscu z dostępem do gniazda kontaktu
2. (punkt opcjonalny) Wkładamy pojemnik do średnicy 60 mm oraz napełniamy go wodą.
3. Podłączamy urządzenie do prądu.
4. Naciskamy przycisk „WŁĄCZ”
5. Na ekranie pokazuje nam się komunikat powitalny
6. Przyciskami „GÓRA”-”DÓŁ” ustawiamy czas
7. Po wybraniu odpowiedniego dla nas czasu naciskamy przycisk „WŁĄCZ”
8. Myjemy zęby.
9. Po zakończeniu mycia, możemy odpiąć urządzenie od prądu.

Dane techniczne urządzenia

3.1 Użyte komponenty

Urządzenie składa się z następujących komponentów:

a) Arduino MEGA 2560 + buzzer



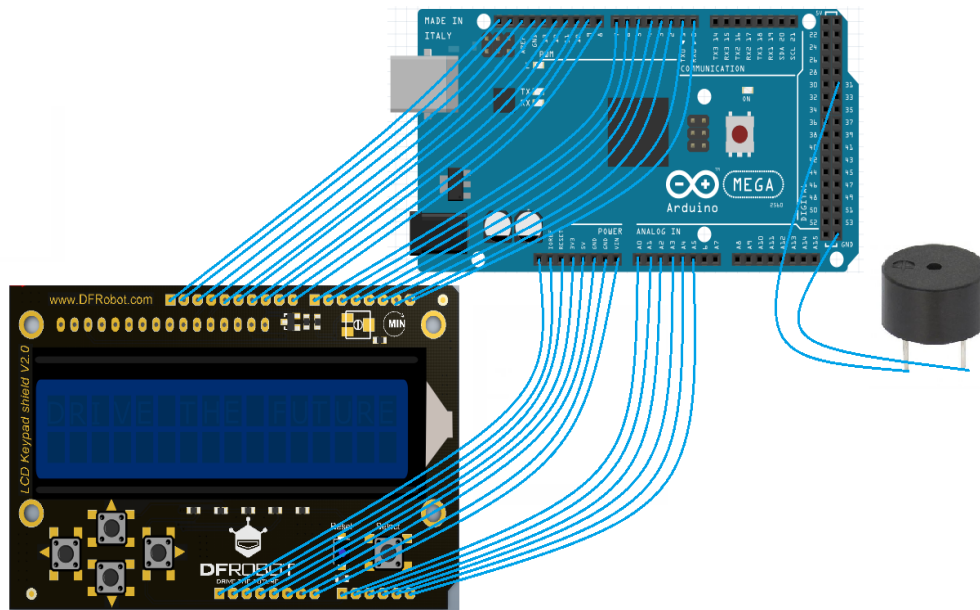
b) LCD keypadShield V2.0



c) Obudowa urządzenia



3.2 Uproszczony rysunek techniczny



3.3 Kod programu

Plik główny:

```
#include "lcdDisplay.h"
```

```
// definition of buttons;
#define btnUP    1
#define btnDOWN  2
#define btnSELECT 4
#define btnNONE  5
```

```
Screen screen; // create an object
```

```
/**Global variables**/
```

```
int flag = 0;
int lcd_key  = 0;
int adc_key_in = 0;
bool welcome = true;
```

```
//read the button value
```

```
int read_LCD_buttons()
{
    adc_key_in = analogRead(0); // read analog A0 value
    if(adc_key_in > 1000) return btnNONE;
    if(adc_key_in < 250) return btnUP;
    if(adc_key_in < 450) return btnDOWN;
    if(adc_key_in < 850) return btnSELECT;
```

```

    return btnNONE;
}

void setup() {
    Serial.begin(9600);
    analogWrite(10, LOW);
}

void loop() {
    lcd_key = read_LCD_buttons();

    //*****Buttons*****
    /**Button "+" **
    switch (lcd_key)
    {
        case btnUP:
        {
            if(flag == 0)
            {
                break;
            }
            else
            {
                int interception = 0;
                screen.resetScreen();
                interception = screen.returnPlusSeconds();
                if(interception<10)
                {
                    screen.showTimeWithSecondsReplacement();
                }
            }
            else
            {
                screen.showTimeScreen();
            }
            delay(100);
        }
        break;
    }

    /**Button "-" **
    case btnDOWN:
    {
        if(flag == 0)
        {
            break;
        }
        else
        {
            int interception = 0;

```

```

    screen.resetScreen();
    interception = screen.returnMinusSeconds();
    if(interception<10)
    {
        screen.showTimeWithSecondsReplacement();
    }
    else
    {
        screen.showTimeScreen();
    }
    delay(100);
}
break;
}

/**button main**
case btnSELECT:
{
    analogWrite(10, 150);
    flag = 1;
    if (welcome==false)
    {
        delay(1000);
        if(analogRead(0)<850)
        {
            analogWrite(10,0);
            delay(1000);
            analogWrite(10,150);

            screen.resetScreen();
            screen.welcomeScreen();
            welcome=true;
            break;
        }
        screen.mainFunctionCoutingDownScreen();
    }
    else
    {
        screen.resetScreen();
        screen.welcomeScreen();
        welcome=false;
        screen.resetScreen();
        screen.showTimeWithSecondsReplacement();
    }
    break;
}
}
}

```

Klasa lcdDisplay:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

```
class Screen{
public:
```

```
int seconds = 0;
int minutes = 2;
int secondsReplacement = 0;
int buzzer = 31;
```

```
void welcomeScreen()
{
    pinMode(buzzer,OUTPUT);
    minutes = 2;
    seconds = 0;
    lcd.begin(16,2);
    lcd.print("Witaj");
    lcd.setCursor(3,1);
    lcd.print("Czas na mycie");
    delay(2000);
}
```

```
void goodbyeScreen()
{
    resetScreen();
    lcd.setCursor(0,0);
    lcd.print("To juz koniec ");
    delay(2000);
    resetScreen();
    lcd.setCursor(0,0);
    lcd.print("Do nastepnego");
    lcd.setCursor(11,1);
    lcd.print("razu!");
    delay(5000);
    minutes=2;
    seconds=0;
}
```

```
void resetScreen()
{
    lcd.clear();
    lcd.setCursor(0,0);
}
```

```
int returnPlusMinutes()
{
    minutes = minutes + 1;
    return minutes;
}
```

```

int returnPlusSeconds()
{

    if(seconds >= 55)
    {
        returnPlusMinutes();
        seconds = -5;
    }
    seconds=seconds+5;
    return seconds;
}

```

```

int returnMinusMinutes()
{
    minutes = minutes -1;
    return minutes;
}

```

```

int returnMinusSeconds()
{
    if(seconds <= 0 && minutes > 0)
    {
        returnMinusMinutes();
        seconds = 60;
    }
    seconds=seconds-5;
    return seconds;
}

```

```

void showTimeScreen()
{
    lcd.print("Twoj czas: ");
    lcd.setCursor(0,1);
    lcd.print("0");
    lcd.setCursor(1,1);
    lcd.print(minutes);
    lcd.setCursor(2,1);
    lcd.print(" min ");
    lcd.setCursor(7,1);
    lcd.print(seconds);
    lcd.setCursor(9,1);
    lcd.print(" s");
    checkTheValue();
}

```

```

void showTimeWithSecondsReplacement()
{

    checkTheValue();
}

```



```

    resetScreen();
    lcd.print("Twoj czas: ");
    lcd.setCursor(0,1);
    lcd.print("0");
    lcd.setCursor(1,1);
    lcd.print(minutes);
    lcd.setCursor(2,1);
    lcd.print(" min ");
    lcd.setCursor(7,1);
    lcd.print("0");
    lcd.setCursor(8,1);
    lcd.print(seconds);
    lcd.setCursor(9,1);
    lcd.print(" s");
}

```

```

void checkTheValue()
{
    if(minutes < 1)
    {
        resetScreen();
        lcd.print("czas nie ponizej");
        lcd.setCursor(0,1);
        lcd.print("1 minuty");
        delay(1000);

        minutes = 1;
        seconds = 0;

        showTimeWithSecondsReplacement();
        delay(100);
    }
}

```

```

if(minutes==5 && seconds >0)
{
    resetScreen();
    lcd.print("nie przekraczac");
    lcd.setCursor(0,1);
    lcd.print("5 minut");
    delay(1000);

    minutes = 5;
    seconds = 0;
    showTimeWithSecondsReplacement();
    delay(100);
}
}

```

```

void mainFunctionCoutingDownScreen()
{
    resetScreen();
    secondsReplacement = seconds;
    int i = 0;

    tone(buzzer,950,1000);
    delay(1000);
    noTone(buzzer);
    analogWrite(10, 150);

    while(secondsReplacement>=0)
    {

        if(analogRead(0)< 850 && analogRead(0)> 650)
        {
            resetScreen();
            while(analogRead(0)< 850 && analogRead(0)> 650)
            {
                pauseScreen();
                delay(1000);
                resetScreen();
            }

        }
        resetScreen();
        countingDownScreen();
        if(secondsReplacement<=0 && minutes>0)
        {
            secondsReplacement=60;
            minutes--;
        }
        if(secondsReplacement<=0 && minutes==0)
        {
            goodbyeScreen();
            showTimeWithSecondsReplacement();
            break;
        }
        if(secondsReplacement<=10 && minutes==0)
        {
            analogWrite(10, 150);
            if(secondsReplacement<=3)
            {
                if(secondsReplacement<=1)
                {

                    tone(buzzer,1200,500);
                    analogWrite(10, 150);
                    delay(100);
                    tone(buzzer,950,250);
                }
            }
        }
    }
}

```

```

        analogWrite(10, 150);
        delay(100);
        noTone(buzzer);
        analogWrite(10, 150);
    }
    else
    {
        tone(buzzer,950,500);
        delay(30);
        noTone(buzzer);
        analogWrite(10, 150);
    }
}
else
{
    tone(buzzer,700,250);
    analogWrite(10, 150);
    delay(30);
    noTone(buzzer);
    analogWrite(10, 150);
}

}
secondsReplacement--;
delay(1000);
}
}

void pauseScreen()
{
    if(secondsReplacement<10)
    {
        lcd.print("zatrzymanie na: ");
        lcd.setCursor(0,1);
        lcd.print("0");
        lcd.setCursor(1,1);
        lcd.print(minutes);
        lcd.setCursor(2,1);
        lcd.print(" min ");
        lcd.setCursor(7,1);
        lcd.print("0");
        lcd.setCursor(8,1);
        lcd.print(secondsReplacement);
        lcd.setCursor(9,1);
        lcd.print(" s");
    }
    else
    {
        lcd.print("zatrzymanie na: ");
        lcd.setCursor(0,1);

```

```

        lcd.print("0");
        lcd.setCursor(1,1);
        lcd.print(minutes);
        lcd.setCursor(2,1);
        lcd.print(" min ");
        lcd.setCursor(7,1);
        lcd.print(secondsReplacement);
        lcd.setCursor(9,1);
        lcd.print(" s");
    }

}

void countingDownScreen()
{
if(secondsReplacement<10)
{
    lcd.print("koniec mycia za: ");
    lcd.setCursor(0,1);
    lcd.print("0");
    lcd.setCursor(1,1);
    lcd.print(minutes);
    lcd.setCursor(2,1);
    lcd.print(" min ");
    lcd.setCursor(7,1);
    lcd.print("0");
    lcd.setCursor(8,1);
    lcd.print(secondsReplacement);
    lcd.setCursor(9,1);
    lcd.print(" s");
}
else
{
    lcd.print("koniec mycia za: ");
    lcd.setCursor(0,1);
    lcd.print("0");
    lcd.setCursor(1,1);
    lcd.print(minutes);
    lcd.setCursor(2,1);
    lcd.print(" min ");
    lcd.setCursor(7,1);
    lcd.print(secondsReplacement);
    lcd.setCursor(9,1);
    lcd.print(" s");
}
}
};

```