

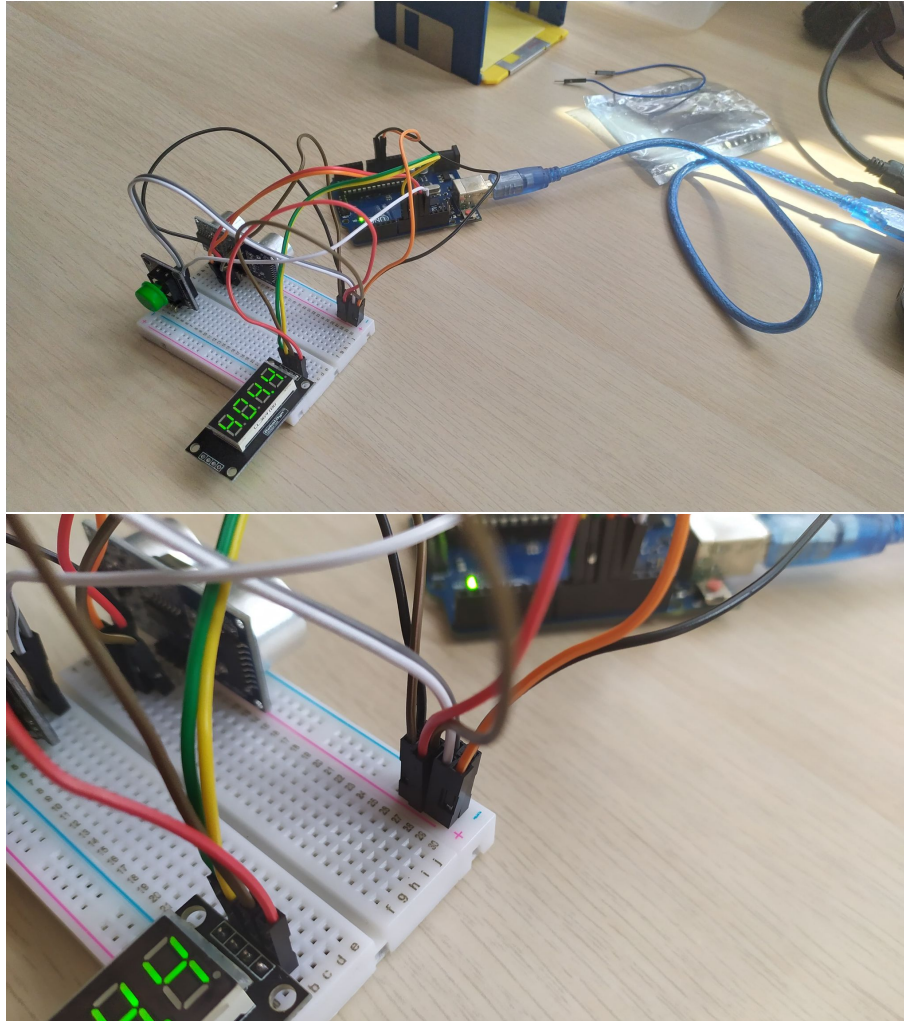
# Ultradźwiękowy miernik odległości

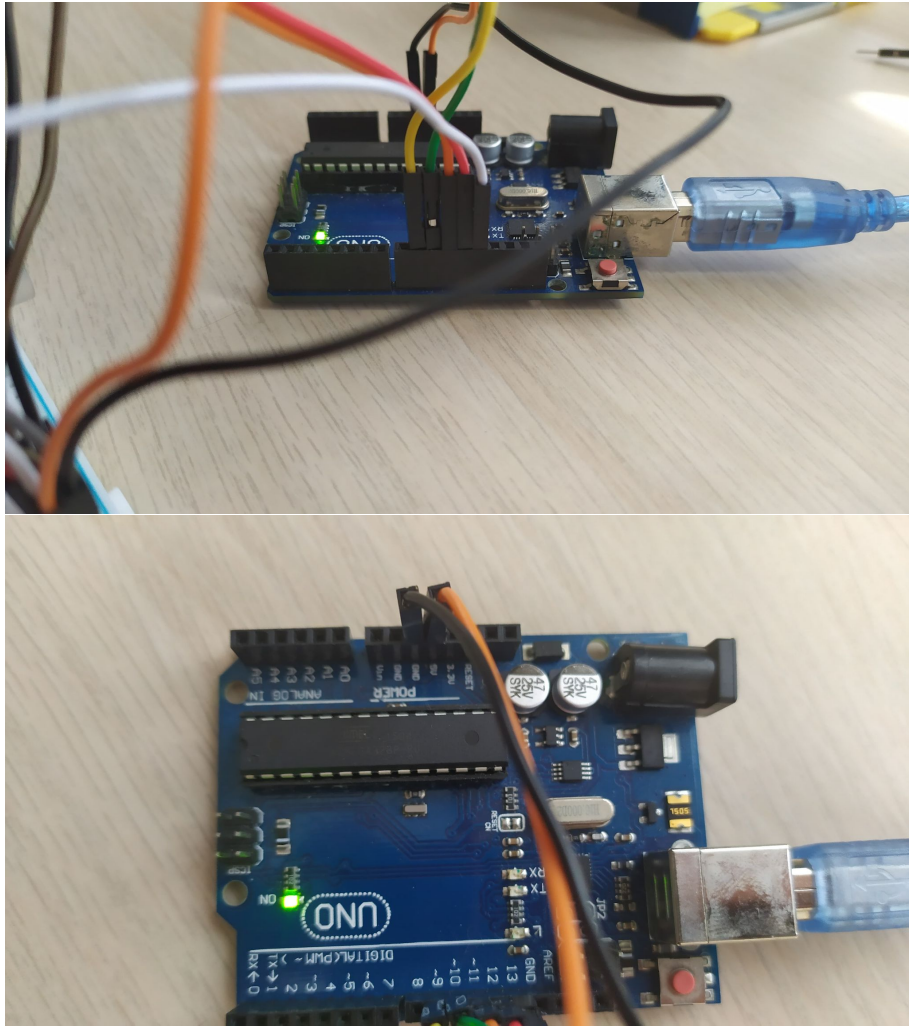
*Denis Lyakhov*  
*Piotr Maliszewski*

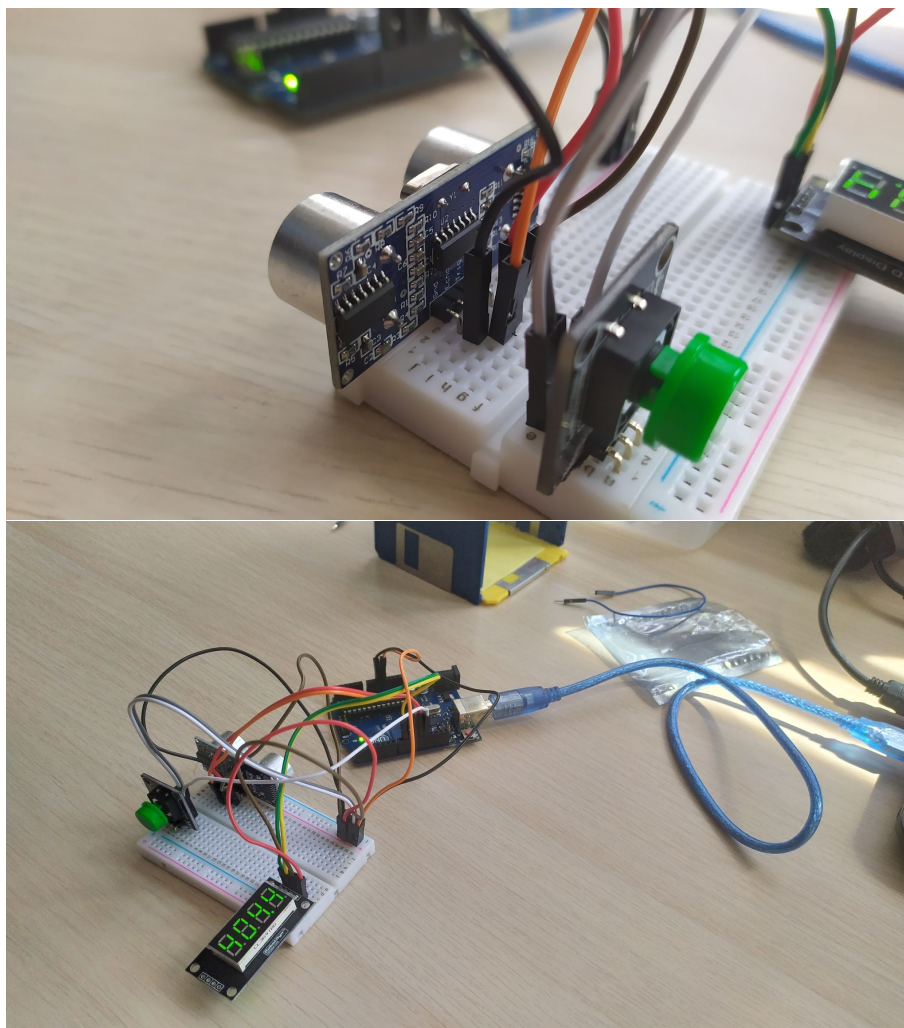
GRUPA II A  
INFORMATYKA ROK II  
2019/2020

2020  
Wrzesień

# 1 Schemat połączeń płytki i dodatkowych modułów







## 2 Opis algorytmu

1. W pętli wyświetlamy migający napis w postaci czterech poziomych kresek:

```
while(true){  
  
    if(time < tickInterval){  
        display.setSegments(defaultDisplay);  
    } else if(time < tickInterval*2){  
        display.clear();  
    }  
}
```

```

        } else {
            time = 0;
        }
    }
}

```

2. **Po naciśnięciu i zwolnieniu przycisku wychodzimy z pętli:**

```

if(digitalRead(buttonPort) == HIGH){
    btnState = 1;
} else if (btnState == 1){
    btnState = 0;
    break;
}

```

3. **Pobieramy 20 pomiarów z czujnika i dodajemy je do zmiennej:**

```

for(int i = 0; i < 20; i++){

    digitalWrite(sensorTrigger , HIGH);
    delay(10);
    digitalWrite(sensorTrigger , LOW);

    mean += pulseIn(sensorEcho , HIGH)*0.034/2;

}

```

4. **Obliczamy średnią na podstawie pomiarów:**

```

mean = mean / 2.0;

```

5. **Wyświetlamy wynik na ekranie:**

```

display.showNumberDecEx(mean, 0x20 , true , 4, 0);

```

6. **Ponowne naciśnięcie i zwolnienie przycisku:**

```

while(true){
    if(digitalRead(buttonPort) == HIGH){
        btnState = 1;
    } else if (btnState == 1){
        break;
    }
}

```

### 3 Elementy programu

- buttonPort – Numer portu wyjścia przycisku
- sensorTrigger – Numer portu do aktywacji czujnika
- sensorEcho – Numer portu do odczytu wyjścia z czujnika
- displayClk – Numer portu CLK wyświetlacza
- displayDio – Numer portu przeznaczonego dla diód w wyświetlaczu
- btnState – Zmienna do zapamiętania poprzedniego stanu przycisku
- tickInterval – Długość wyświetlania (ilość iteracji w pętli)
- time – Bierząca iteracja pętli
- mean – Średnia pomiarów

### 4 Kod źródłowy

```
#include <TM1637Display.h>

const int buttonPort = 13;
const int sensorTrigger = 12;
const int sensorEcho = 11;
const int displayClk = 10;
const int displayDio = 9;

const int tickInterval = 35;

int btnState, time = 0;

TM1637Display display(displayClk, displayDio);

const uint8_t defaultDisplay[] = {
    SEG_G,
    SEG_G,
    SEG_G,
    SEG_G
};

void displayDefaultState(){
    time = 0;
```

```

btnState = 0;

while(true){

    if(time < tickInterval){
        display.setSegments(defaultDisplay);
    } else if(time < tickInterval*2){
        display.clear();
    } else {
        time = 0;
    }

    if(digitalRead(buttonPort) == HIGH){
        btnState = 1;
    } else if (btnState == 1){
        btnState = 0;
        break;
    }

    time++;

}

}

void setup() {

    Serial.begin(9600);

    Serial.print("Starting configuration");

    pinMode(buttonPort , INPUT);
    pinMode(sensorEcho , INPUT);

    display.setBrightness(0x0a);

    pinMode(sensorTrigger , OUTPUT);
    digitalWrite(sensorTrigger , LOW);

}

void loop() {

    displayDefaultState();

    float mean = 0;

```

```

for(int i = 0; i < 20; i++){

    digitalWrite(sensorTrigger , HIGH);
    delay(10);
    digitalWrite(sensorTrigger , LOW);

    mean += pulseIn(sensorEcho , HIGH)*0.034/2;

}

mean = mean / 2.0;

display.showNumberDecEx(mean, 0x20, true, 4, 0);

delay(100);

while(true){
    if(digitalRead(buttonPort) == HIGH){
        btnState = 1;
    } else if (btnState == 1){
        break;
    }
}

delay(1);

}

```

## 5 Biblioteka TM1637Display.h

Główną część wyświetlania stanowi następująca funkcja:

*showNumberDecEx(number, dots, leading\_zeros, length, position)*

Funkcja ta pozwala ustalić, gdzie ma być wyświetlony przecinek. Drugi argument pozwala na spracyzowanie miejsca przecinka.

### Przykładowo:

- 0b10000000 – 0.000
- 0b01000000 – 00.00
- 0b00100000 – 000.0
- 0b11100000 – 0.0.0.0

### Inne funkcje zawarte w bibliotece:

- setSegments() — Ustawia daną wartość na segmentach



- `showNumberDec()` — Wyświetla liczbę w postaci dziesiętnej
- `showNumberDecEx()` -- Wyświetla liczbę w postaci dziesiętnej z przecinkiem
- `setBrightness()` -- Ustawia jasność wyświetlacza
- `clear()` — Czyści wyświetlacz

**Korzystaliśmy z:**

Poradnik: <https://www.makerguides.com/tm1637-arduino-tutorial/>

Link do biblioteki: <https://github.com/avishorp/TM1637/tree/master/docs>