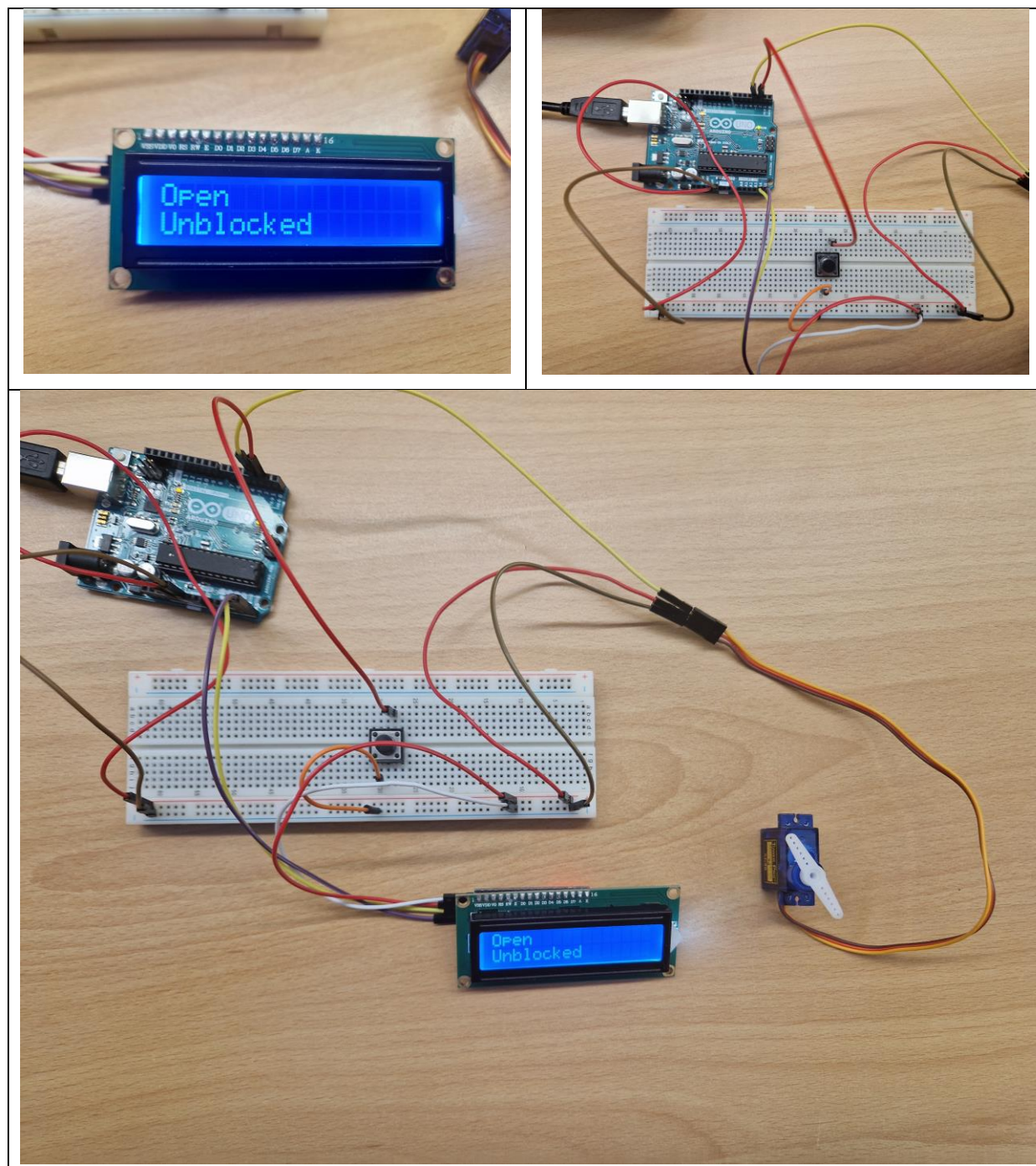


[Z3] Arduino UNO – elementy wykonawcze

Grupa laboratoryjna: L1 Podgrupa: 2	Arkadiusz Michowski (127 280)	
	Mikołaj Lebioda (101 507)	
	Prowadzący zajęcia:	dr inż. Ariel Antonowicz

ZDJĘCIA ZREALIZOWANEGO UKŁADU



KOD ZREALIZOWANEGO ALGORYTMU (wraz z komentarzami)

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <Servo.h>

#define WIRE Wire

// Piny
#define PIN_BUTTON 2
#define PIN_SERVO 3

// Interfejs do LCD, pierwszy parametr to znaleziony wcześniej adres
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Stany: 1)Zablokowany/odblokowany 2)Otwarty/zamknięty
bool isBlocked = true;
bool isOpened = false;

// Funkcja przerwania musi zawierać krótką instrukcję
// Dlatego będziemy używać "jednorazowej" flagi,
// która zapamięta, że w pętli loop trzeba pokazać obecny stan na LCD
bool shouldPrintCurrentState = false;

// Interfejs do silnika, pozycja oraz szybkość
Servo servo;
int position = 0;
int speed = 6;

void setup() {
    WIRE.begin();
    Serial.begin(9600);

    pinMode(PIN_BUTTON, INPUT_PULLUP);

    // Zresetuj pozycję silnika
    servo.attach(PIN_SERVO);
    servo.write(0);

    // Zainicjalizuj LCD
    lcd.init();
    lcd.backlight();
    printCurrentState();

    // Przypisanie funkcji handleButton do przycisku (przerwanie)
    attachInterrupt(digitalPinToInterrupt(PIN_BUTTON), handleButton, FALLING);
}

// Wyświetl tymczasową wiadomość na 3 sekundy
void printTemporaryMessage(String line) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(line);

    // Pamiętaj, by potem wrócić do wyświetlania aktualnego stanu
    shouldPrintCurrentState = true;
}
```

```

    delay(3000);
}

// Wyświetlanie stanu: otwarty/zamknięty oraz zablokowany/odblokowany
void printCurrentState() {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(isOpened ? "Open" : "Closed");
    lcd.setCursor(0,1);
    lcd.print(isBlocked ? "Blocked" : "Unblocked");
}

// Zmień stan blokady na przeciwny
void handleButton() {
    isBlocked = !isBlocked;

    Serial.println("button clicked");

    // Pamiętaj, by potem zaktualizować LCD! (za długi czas wywołania na użycie
    tutaj, w funkcji przerwania)
    shouldPrintCurrentState = true;
}

// Otwórz/zamknij
void open(bool newOpened) {
    if (isBlocked) {
        Serial.println("I am blocked!");
        return;
    }

    // Wiadomość tymczasowa jeśli nowy stan jest taki sam jak stary
    String msg = "";
    if (isOpened && newOpened) msg = "Already open";
    if (!isOpened && !newOpened) msg = "Already closed";

    isOpened = newOpened;

    // Pamiętaj, by wyświetlić nowy stan
    shouldPrintCurrentState = true;

    // Zmień pozycję silnika w zależności od stanu
    servo.write(isOpened ? 90 : 0);

    // Jeśli była jakaś tymczasowa wiadomość, wyświetl
    if (msg.length()) printTemporaryMessage(msg);
}

void loop() {
    // Sprawdź, czy powinniśmy wyświetlić obecny stan na LCD
    if (shouldPrintCurrentState) {
        printCurrentState();

        shouldPrintCurrentState = false;
    }

    // Odczytaj ciąg znaków z wejścia i obsłuż jako komendę
    if (Serial.available()) {

```

```
String str = Serial.readStringUntil("\n");
str.trim();

if      (str == "o") open(true);
else if (str == "c") open(false);
else      printTemporaryMessage("wrong cmd");
}
}
```