**Spis treści**

1. Schematy ideowe 2

1.1. Podłączenie diody do wyjścia cyfrowego 2

1.2. Sterowanie silnikiem przy użyciu potencjometru 2

1.3. Podłączenie jednego czujnika temperatury 3

1.4. Podłączenie jednego czujnika temperatury w trybie pasożytniczym 3

1.5. Podłączenie wielu czujników temperatury 4

2. Code 5

2.1. Mruganie diodą 5

2.2. Zapis do SerialPort 5

2.3. Odczyt z SerialPort 5

2.4. Sterowanie PWM 6

2.5. Sterowanie PWM przy użyciu SerialPortu 6

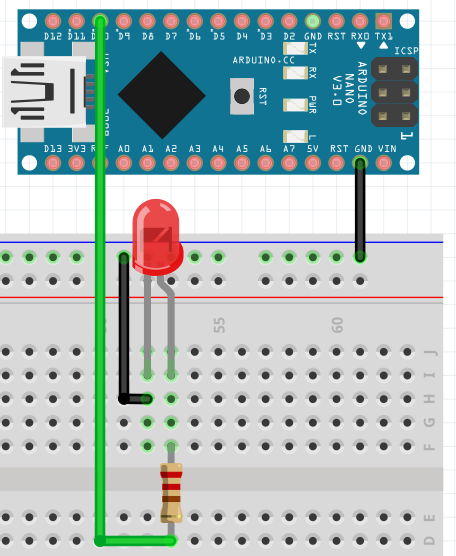
2.6. Sterowanie PWM przy użyciu potencjometru 7

2.7. Odczyt urządzeń z linii OneWire 8

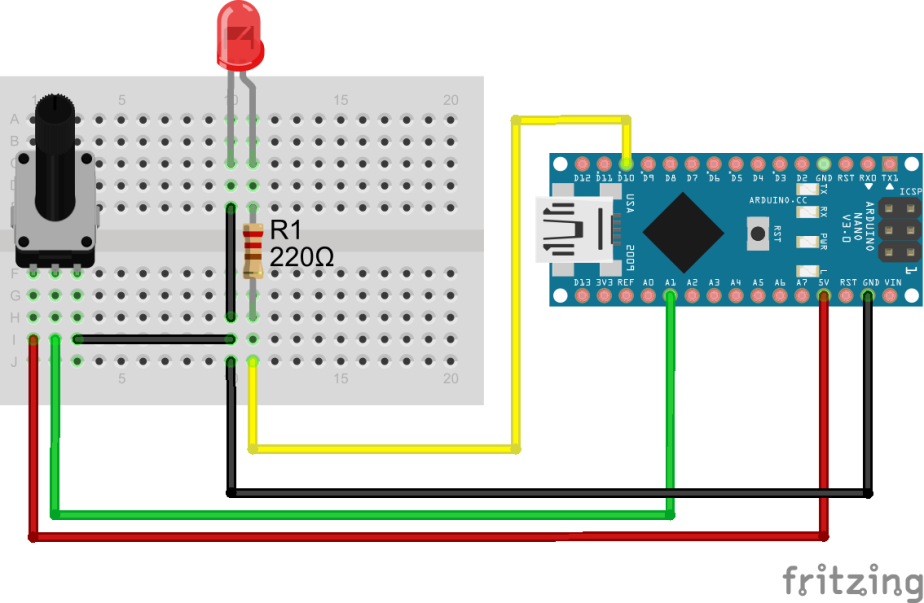
2.8. Odczyt temperatury 9

# Schematy ideowe

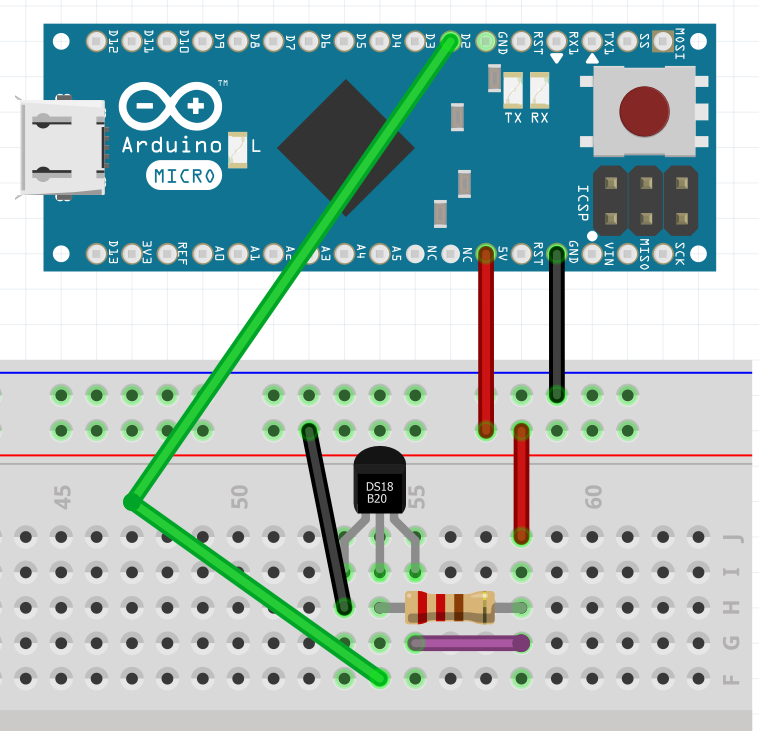
## Podłączenie diody do wyjścia cyfrowego



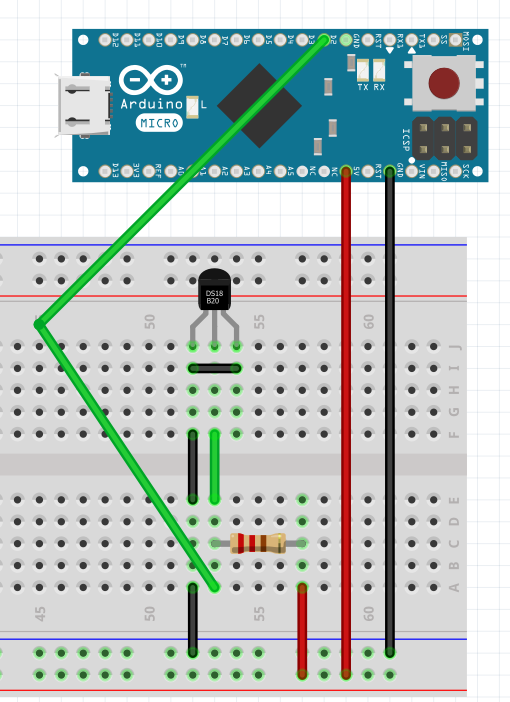
## Sterowanie silnikiem przy użyciu potencjometru



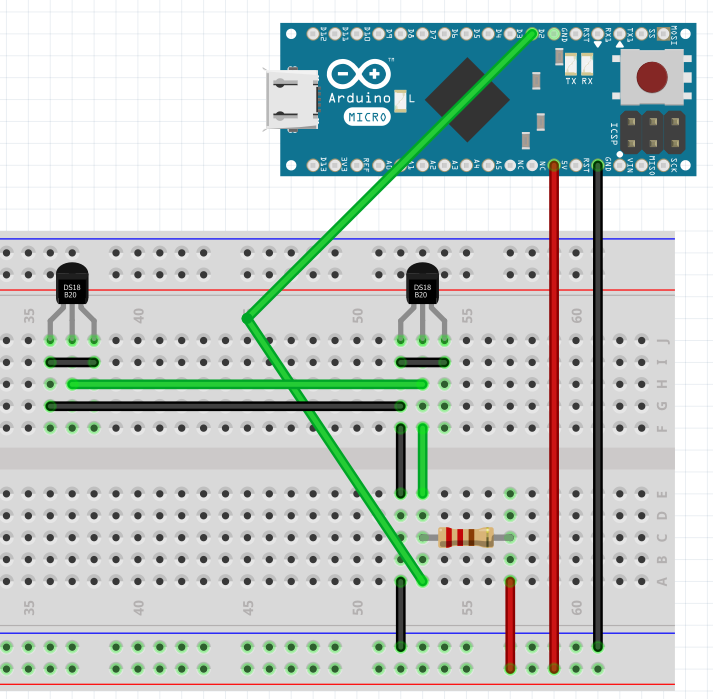
## Podłączenie jednego czujnika temperatury



## Podłączenie jednego czujnika temperatury w trybie pasożytniczym



## Podłączenie wielu czujników temperatury



# Code

## Mruganie diodą

|  |
| --- |
| **int** portNumber = 10;  **void** setup() {  pinMode(portNumber, OUTPUT);  }  **void** loop() {  digitalWrite(portNumber, HIGH);  delay(1000);  digitalWrite(portNumber, LOW);  delay(1000);  } |

## Zapis do SerialPort

|  |
| --- |
| **int** iterator = 0;  **void** setup(){  Serial.begin(9600);  }  **void** loop(){  iterator = iterator + 1;  Serial.print("Iteration : "); //send the string “hello” and return the length of the string.  Serial.println(iterator);  delay(1000);  } |

## Odczyt z SerialPort

|  |
| --- |
| // whole content  String incomingContent = "";  // single character  **char** character;  **void** setup() {  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps  Serial.println("Please insert value");  }  **void** loop() {    **while** (Serial.available()) {  character = Serial.read();  incomingContent.concat(character);  }  **if** (incomingContent != "") {  Serial.println(incomingContent);  incomingContent = "";  }  } |

## Sterowanie PWM

|  |
| --- |
| **int** pwmPort = 6;  **int** pwmValueLow = 100;  **int** pwmValueHigh = 255;  **void** setup() {  // SetUp proper pinmode  pinMode(pwmPort, OUTPUT);  }  **void** loop() {  // write proper value to PWM output  analogWrite(pwmPort, pwmValueLow);  //Delay  delay(3000);  // write proper value to PWM output  analogWrite(pwmPort, pwmValueHigh);  //Delay  delay(3000);  } |

## Sterowanie PWM przy użyciu SerialPortu

|  |
| --- |
| // PWM properties  **int** pwmPort = 6;  **int** pwmValueLow = 100;  **int** pwmValueHigh = 255;  // whole content  String incomingContent = "";  // single character  **char** character;  **void** setup() {  // SetUp proper pinmode  pinMode(pwmPort, OUTPUT);  // opens serial port, sets data rate to 9600 bps  Serial.begin(9600);  Serial.println("Please insert value");  }  **void** loop() {  **while** (Serial.available()) {  character = Serial.read();  incomingContent.concat(character);  }  **if** (incomingContent != "" && incomingContent.length() == 2) {  **if** (validateContent(incomingContent)) {  Serial.println("Recieved valid percentage value");  analogWrite(pwmPort, convertPercentageToPWM(incomingContent));  } **else** {  Serial.println("Something went wrong");  analogWrite(pwmPort, pwmValueLow);  }  incomingContent = "";  }  }  **boolean** validateContent(String incomingContent) {  Serial.print("Recieved content:");  Serial.println(incomingContent);  // always = 2  **int** stringLength = incomingContent.length();  **boolean** result = **true**;  **for** (**int** i = 0; i < stringLength; i++) {  **char** singleChar = incomingContent.charAt(i);  result &= isDigit(singleChar);  **if** (!result) {  Serial.println("Recieved non digital character");  **return** result;  }  }  **return** result;  }  **int** convertPercentageToPWM(String incomingContent) {  **float** percentage = incomingContent.toFloat();  Serial.print("Percentage: ");  Serial.println(percentage);  **float** result = (percentage / 100) \* pwmValueHigh;  Serial.print("Writing to PWM: ");  Serial.println((**int**) result);  **return** (**int**) result;  } |

## Sterowanie PWM przy użyciu potencjometru

|  |
| --- |
| Sint inputPin = A0; // set input pin for the potentiometer  **int** inputValue = 0; // potentiometer input variable  **int** ledPin = 6; // set output pin for the LED  **void** setup() {  // declare the ledPin as an OUTPUT:  pinMode(ledPin, OUTPUT);  // opens serial port, sets data rate to 9600 bps  Serial.begin(9600);  }  **void** loop() {  // read the value from the potentiometer:  inputValue = analogRead(inputPin);  Serial.print("Potentiometer value: ");  Serial.println(inputValue);  // send the square wave signal to the LED:  analogWrite(ledPin, inputValue / 4);  delay(500);  } |

## Odczyt urządzeń z linii OneWire

|  |
| --- |
| #include <OneWire.h>  // sensor Port  **const** **byte** ONEWIRE\_PIN = 2;  OneWire onewire( ONEWIRE\_PIN);  **void** setup() {  **while** (!Serial)  ;  Serial.begin(9600);  }  **void** loop() {  **byte** address[8];  onewire.reset\_search();  **while** (onewire.search(address)) {  **if** (address[0] != 0x28)  **continue**;  // something went wrong  **if** (OneWire::crc8(address, 7) != address[7]) {  Serial.println("Wrong address");  **break**;  }  // print address starting with Ox  **for** (**byte** i = 0; i < 8; i++) {  Serial.print(F("0x"));  Serial.print(address[i], HEX);  **if** (i < 7)  Serial.print(", ");  }  Serial.println();  }  **while** (1) {  // "ugly" loop which avoids reading sensors all over again  }  } |

## Odczyt temperatury

|  |
| --- |
| #include <OneWire.h>  // sensor Port  **const** **byte** ONEWIRE\_PIN = 2;  OneWire onewire( ONEWIRE\_PIN);  **void** setup() {  **while** (!Serial)  ;  Serial.begin(9600);  }  **void** loop() {  **byte** address[8];  onewire.reset\_search();  **while** (onewire.search(address)) {  **if** (address[0] != 0x28)  **continue**;  // something went wrong  **if** (OneWire::crc8(address, 7) != address[7]) {  Serial.println("Wrong address");  **break**;  }  // print address starting with Ox  **for** (**byte** i = 0; i < 8; i++) {  Serial.print(F("0x"));  Serial.print(address[i], HEX);  **if** (i < 7)  Serial.print(", ");  }  Serial.println();  }  **while** (1) {  // "ugly" loop which avoids reading sensors all over again  } |