

Piotr KISIEL 

ORCID: 0000-0001-9680-8976, Dr inż., Państwowa Wyższa Szkoła Wschodnioeuropejska w Przemysłu, ul. Książąt Lubomirskich 6; 37-700 Przemysł, I Liceum Ogólnokształcące im. Juliusza Słowackiego w Przemysłu, e-mail: piotrkisiel@wp.pl

PROGRAMOWANIE MIKROKONTROLERÓW W SZKOLE ŚREDNIEJ Z UŻYCIEM PLATFORMY ARDUINO

THE NUTS AND BOLTS OF PROGRAMMING SCIENCE IN SECONDARY SCHOOL USING THE ARDUINO FRAMEWORK

Słowa kluczowe: nauka programowania mikrokontrolerów, Arduino, informatyka szkoła średnia.

Keywords: learning programming microcontrollers, Arduino, computer science in secondary school.

Streszczenie

Umiejętność doboru odpowiedniego środowiska informatycznego w realizacji rozwiązań problemów jest kluczową umiejętnością współczesnego programisty. Obecne możliwości łączenia mikroelektroniki i programowania na tanich i ogólnodostępnych platformach, są olbrzymią szansą uatrakcyjnienia procesu edukacyjnego nauki programowania. Niniejszy artykuł porusza praktyczne aspekty programowania mikrokontrolerów Arduino na poziomie szkoły średniej.

Abstract

Right selection of IT environment to implement solutions to problems is a fundamental ability of modern programmer. The current possibilities of combining microelectronics and programming on low-cost and generally available platforms are a huge opportunity to make much more attractive the learning process of programming. This article deals with practical aspects of programming Arduino microcontrollers at the high school level.

Wstęp

Celem niniejszego opracowania jest przybliżenie zmian wymuszonych przez nową podstawę programową obejmującą zagadnienia nauczania informatyki w szkole średniej. Potrzeba badania form nauczania w praktycznym ujęciu reali-

zacji poszczególnych zagadnień, jest kluczowa, mając na uwadze fakt szczątkowych szkoleń nauczycieli z nowej podstawy programowej. Rok szkolny 2019/2020 będzie rokiem znamionym dla edukacji w Polsce z racji powrotu 4-letnich liceów i 5-letnich techników zawodowych. Zmianie ulega zakres nauczania w omawianym przedmiocie, z jakim muszą uporać się nauczyciele. Nie jest to jedynie zmiana ilościowa, a również jakościowa w stosunku do treści zawartych w dotychczasowych programach nauczania.

Bardzo duże zmiany zarówno pod względem dydaktycznym, jak i merytorycznym zawiera w sobie program dedykowany dla przedmiotu informatyka, zarówno na poziomie podstawowym, jak i rozszerzonym. W pewnym stopniu jest to również zmiana wymuszona błyskawicznym rozwojem tej dyscypliny nauki i techniki w otaczającym nas świecie.

Analiza zapisów treści nauczania nowej podstawy programowej

Poniżej jest przytoczony fragment podstawy programowej dla liceum, technikum i branżowej szkoły II stopnia obowiązujący od roku szkolnego 2019/2020 w klasie pierwszej.

INFORMATYKA ZAKRES PODSTAWOWY I ROZSZERZONY

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Zakres podstawowy.

Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych. W szczególności programuje algorytmy z punktu I.2);

2) do realizacji rozwiązań problemów dobiera odpowiednie środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

*3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
a) projektuje modele dwuwymiarowe i trójwymiarowe, tworzy i edytuje projekty w grafice rastrowej i wektorowej, wykorzystuje różne formaty obrazów, przekształca pliki graficzne uwzględniając wielkość i jakość obrazów....*

Z satysfakcją należy odnotować pojawienie się w podstawie programowej postulowanych na łamach wielu artykułów, treści edukacyjnych, związanych z grafiką komputerową¹ oraz modelowaniem przestrzennym w nauczaniu informatyki².

¹ P. Kisiel, *Grafika komputerowa i informatyczne modelowanie struktur przestrzennych w programie kształcenia liceum ogólnokształcącego*, „Dydaktyka Informatyki” 2017, s. 242–249.

² P. Kisiel, *Praktyczne aspekty nauki programowania w szkole średniej*, „Dydaktyka Informatyki” 2018, s. 147–152.

W obecnym artykule badane są praktyczne formy realizacji następującego zapisu podstawy programowej:

2) do realizacji rozwiązań problemów dobiera odpowiednie środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki.

Pozytywnie należy ocenić zapis ustawodawcy o różnorodnych środowiskach informatycznych. Do tej pory bowiem, nauka informatyki, zwłaszcza na poziomie podstawowym, oscylowała pomiędzy klawiaturą a ekranem monitora, zawężając proces edukacyjny do mało interesujących, coraz bardziej odstających od zmieniającej się rzeczywistości zagadnień. Współcześnie elementy informatyki odnajdziemy już w niemal każdej dziedzinie działalności ludzkiej i znakiem czasu jest omówienie na poziomie szkoły, specyfiki informatyki w szerszym zakresie. Przełomowym jest zapis o stosowaniu informatyki w elementach robotyki. Obecnie jednak, na poziomie szkoły średniej potrzeba czegoś więcej niż przetarte, a wręcz przepracowane rozwiązania stosowania „klocko-robotyków”³ i stereotypów pt. „ramię robota, walki robotów” itd.

Platforma Arduino w procesie dydaktycznym

Wychodząc naprzeciw nowej podstawie programowej badano możliwości zastosowania platformy programistycznej Arduino w procesie dydaktycznym. Arduino bowiem otwiera nowe możliwości stosowania robotyki, a zarazem zaawansowanych układów elektronicznych.

Celem projektu Arduino jest przygotowanie narzędzi – ogólnodostępnych, tanich, niewymagających dużych nakładów finansowych, elastycznych i łatwych w użyciu przez hobbystów. Platforma Arduino stanowi również alternatywę dla osób, które nie mają dostępu do bardziej zaawansowanych kontrolerów, wymagających bardziej skomplikowanych narzędzi⁴. Język programowania Arduino jest oparty na środowisku Wiring i zasadniczo na języku C/C++⁵, stąd też następuje naturalne przejście od teoretycznej nauki programowania do programowania układów zewnętrznych, poza komputerem PC.

Szczególnie obejrzenie efektów swojej pracy programistycznej poza monitorem komputera jest nader atrakcyjną formą dla współczesnego ucznia. Należy pamiętać o tym aspekcie, dobierając sekwencję treści w nauczaniu programowania.

³ LEGO MINDSTORMS, JimuKarbot UBTECH, RoboKidsRoborobo i wiele rozwiązań tego typu z powodzeniem stosowanych jest obecnie w szkołach podstawowych.

⁴ <http://www.arduino.cc/en/Main/Education> (dostęp: 17.03. 2019 r.).

⁵ http://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a (dostęp: 20.02.2019 r.).

Kółko informatyczne

Chcąc przebadać formy kształcenia pod przyszłe lekcje informatyki, nowe treści realizowane były w roku szkolnym 2018/2019 na nadobowiązkowym kółku informatyki, które ze względu na omawiane zagadnienia przybrało charakter nieco mechatroniczny. Z całą bezwzględnością trzeba stwierdzić, iż o badaniach naukowych nie może być tu mowy. Grupa w liczbie ok. ośmiu osób w żaden sposób nie jest grupą reprezentatywną, a jakiegokolwiek zestawienia statystyczne po prostu na tym etapie nie mają najmniejszego sensu. Co zatem było badane? Zważywszy na omawiane zagadnienia, początkowo pojawiły się obawy co do stanu wiedzy z podstaw elektroniki u młodych adeptów programowania kontrolerów. Pamiętać należy, że badana grupa to uczniowie liceum ogólnokształcącego, a nie technikum elektronicznego.

Już po kilku spotkaniach okazało się jednak, że gros zagadnień omawianych było na lekcji fizyki, a solidne podwaliny teoretyczne dały możliwość rozwinięcia i realizacji budowy prostych układów elektronicznych w rzeczywistości.

W praktycznej nauce programowania kontrolerów postawiono w myśl nowej podstawy programowej, na odniesienie się do rzeczywistych zadań i problemów.

W pierwszej kolejności nacisk został położony na pomiar jednej z wielkości fizycznych, jaką interesują się młodzi ludzie – temperatury. Pomimo znacznie większego zainteresowania strefą mediów społecznościowych, temperatura otoczenia z prozaicznych względów pozostaje w kręgu ścisłego zainteresowania młodych ludzi. Dobór właściwego ćwiczenia, to bardzo ważny aspekt z racji przyciągnięcia uwagi i nadania sensu praktycznego, podczas rozwiązywania problemu. Na tym poziomie edukacji, akademickie, teoretyczne wywody i badanie abstrakcyjnych dla młodego człowieka zagadnień mija się z celem, a efekt może być zupełnie przeciwny do zamierzonego.

Realizacje praktyczne

Chcąc ocenić, jak praktyczne aspekty realizacji problemów wpływają na umiejętności uczniów, zostaną omówione podjęte działania.

Do pomiaru wspomnianej temperatury otoczenia, wybrany został układ DS18B20 ze względu na swoje właściwości i charakterystykę.

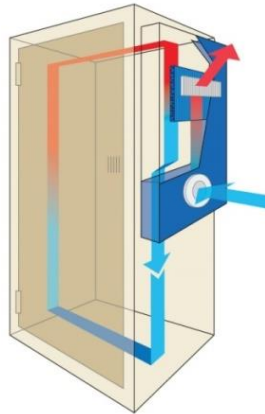
DS18B20 to cyfrowy czujnik temperatury firmy Dallas. Wysyła dane o temperaturze wykorzystując do tego tylko jedno wejście cyfrowe układu Arduino i specjalny protokół o nazwie 1-Wire. Do tego samego wejścia układu można podłączyć wiele czujników. Czujnik podaje sprzętowo temperaturę w stopniach Celsjusza⁶. Pod względem metodycznym jest to wspaniały przykład. Uczeń może

⁶ http://akademia.nettigo.pl/ds18b20/#co_to_jest_ds18b20

zagłębić się w zagadnienia rozdzielczości, dokładności pomiaru, a w końcu podłączenia kilku czujników i odczytu z nich temperatury, dzięki niepowtarzalnym adresom fabrycznie przypisanym do danego egzemplarza. W międzyczasie omówić należy zagadnienie niestandardowych bibliotek łączących warstwę sprzętową urządzeń z programem. W tym przypadku była to biblioteka *OneWire.h*.

W dodatku *appendix 1* przedstawiony jest listing kodu odczytującego numery seryjne czujników. Kod wyświetla je w monitorze portu szeregowego. Ćwiczenie to, stanowiące wprowadzenie do programowania kontrolerów, nie nastęczało uczniom problemów. Wart uwagi jest fakt operowania zmienną, której adres pochodzi z konkretnego fizycznego czujnika. Korelacja tych dwóch zależności stanowi pierwsze otwarcie na komunikację programowo-sprzętową.

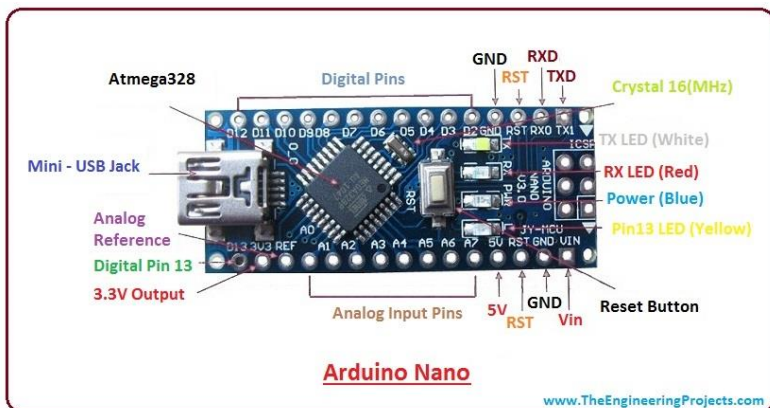
Po kilku lekcjach nadszedł czas na zmierzenie się z rozwiązaniem konkretnego problemu – układu chłodzenia szafy z serwerem. Jest to problem bardzo często spotykany w praktyce, tym bardziej jego rozwiązanie ma wymierne zastosowanie.



Rys. 1. Wymiennik ciepła typu powietrze – powietrze do chłodzenia szafy w obiegu wewnętrznym

Źródło: materiały firmy ASTAT.

Do osiągnięcia optymalnej temperatury wewnątrz szafy serwera nie wystarczy sama konwekcja powietrza. Wymagane jest zastosowanie wentylatora i śluzu powietrza. W projekcie zostały zatem zastosowane dwa termometry cyfrowe DS18B20, wewnętrzny i zewnętrzny, w oparciu o które została zaprogramowana logika obsługująca przełącznik wentylatora i serwo otwierające i zamykające dopływ chłodnego powietrza do szafy. Całość projektu została oparta o sterownik Arduino Nano. O wyborze zdecydowały niewielkie rozmiary, pobór prądu i duże możliwości obsługi wejść i wyjść analogowych oraz cyfrowych.



Rys. 2. Arduino Nano opis pinów

Źródło: www.TheEngineeringProjects.com.



Rys. 3. Projekt sterownika z docelowymi przyłączami oraz służą powietrzną

W omawianym projekcie czujniki temperatury zostały połączone z portem nr 2, portem przekaźnika sterującego wentylatorem stał się port nr 3, natomiast serwomechanizmysterowany został portem nr 5. Początkowo układ został połączony na płytce prototypowej. Rozpoczęło się badanie i testowanie działania urządzenia przez uczniów. Poprawne działanie urządzenia prototypowego pozwoliło na montaż układu na złączu adaptacyjnym dla układów Arduino Nano. Po połączeniu przewodów zewnętrznych okazało się, iż ich długość ma bezpośredni wpływ na ilość zakłóceń generowanych przez serwer, a co za tym idzie –

na stabilność działania całego układu. Zastosowane zostały więc inne rodzaje przewodów, w tym skrętka UTP, bardziej odporna na zakłócenia zewnętrzne, jak też niezbędne modyfikacje samego kodu. Finalnie wszystkie problemy zostały rozwiązane, a projekt okazał się bardzo stabilnie działającym rozwiązaniem. Praktyczna realizacja zadania wymusiła zmianę kodu programowanego sterownika, a tym samym – wymusiła kreatywne myślenie o problemie wśród uczniów. Zadanie to pokazało również, w jaki sposób styczność uczniów z realnymi zadaniami wpływa na sposób i podejście do praktycznego programowania, a w szczególności do programowania mikrokontrolerów.

Appendix 2 zawiera listing kodu opisywanego projektu.

Licznik Geigera

Innym przykładem wykorzystującym platformę programowania Arduino był autorski projekt Przemysława Kinasza z I Liceum Ogólnokształcącego w Przemysłu. Zasada działania licznika Geigera-Müllera opartego o tubę wypełnioną argonem (Ar) z niewielkim dodatkiem par alkoholu znana jest od 1928 r. Na przełomie lat powstało wiele projektów amatorskich, które stały się inspiracją do podjęcia tego tematu.

Tym razem na podstawie materiału dostępnego w sieci internet i literaturze przedmiotowej, powstał projekt oparty o mikrokontroler Arduino Nano. Układ elektroniczny funkcjonuje tutaj jako moduł zliczający impulsy i konwertujący tę liczbę na wartość dawki promieniowania pochłoniętego.

Układ w czasie rzeczywistym oblicza przybliżoną liczbę impulsów na minutę (CPM), a następnie przelicza na wartość dawki promieniowania wyrażoną w $\mu\text{S/h}$.

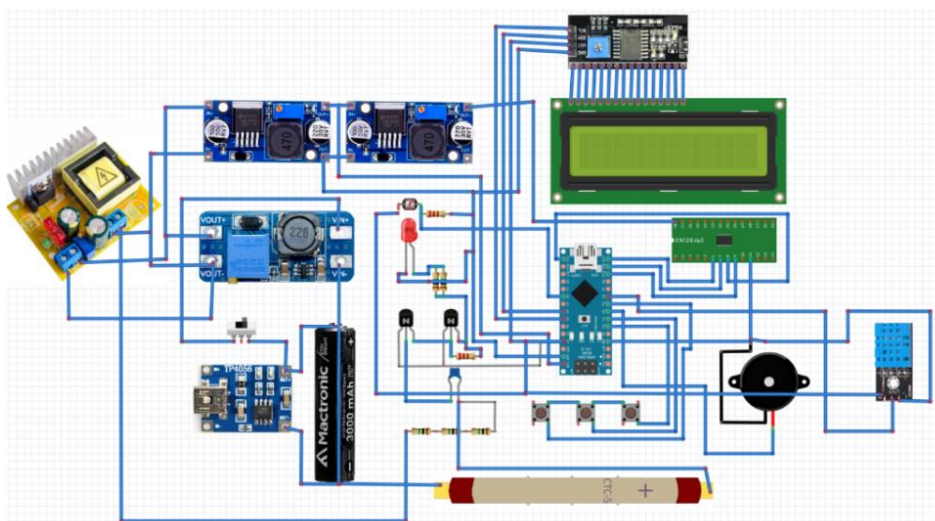
$$\text{CPM} * \text{współczynnik konwersji} = \mu\text{Sv/h}$$

Układ detekcji promieniowania jonizującego, oparto o tubę G-M STS-5 i przetwornicę impulsową podwyższającą napięcie z wejściowego (5–12 V) do 400 V.

Jako urządzenie wyjściowe zastosowany został popularny wyświetlacz ciekłokrystaliczny. Urządzenie wyposażono też w możliwość przesyłania danych za pomocą modułu Ethernet ENC28J60 do sieci.

Wśród funkcjonalności projektu należy odnotować możliwość pomiaru temperatury i wilgotności otoczenia zewnętrznego, czyniąc zeń praprzodka „trikodera”⁷.

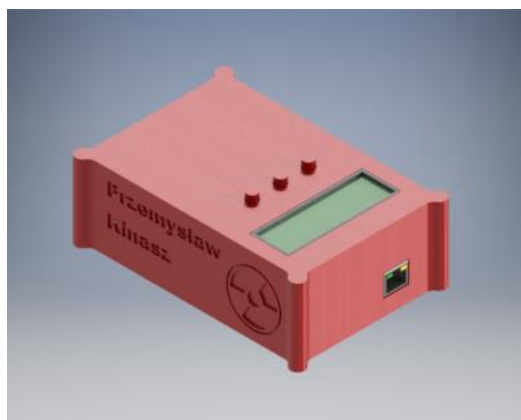
⁷ W fikcyjnym świecie Star Trek Triokorder jest wielofunkcyjnym urządzeniem ręcznym używanym do skanowania środowiska, analizy danych i ich przetwarzania.



**Rys. 4. Licznik Geigera-Müllera z mikrokontrolerem Arduino Nano.
Projekt Przemysław Kinasz**

Appendix 3 zawiera pełny listing kodu omawianego projektu.

Całość urządzenia została umieszczona w obudowie zaprojektowanej i wydrukowanej na drukarce 3D. Warto tutaj odnotować, że drukarka nie była żadnym firmowym rozwiązaniem, a projektem własnej konstrukcji.



Rys. 5. Obudowa dozymetru. Projekt i wykonanie Przemysław Kinasz

Projekt zdobył I miejsce w II Konkursie Twórczości Technicznej pt. „Projekty prosto z garażu” zorganizowanym przez Instytut Nauk Technicznych PWSW w Przemyślu.



Rys. 6. Przemysław Kinasz podczas pracy i prezentacji projektu

Zakończenie

Prezentowane przykłady, realizowane przez uczniów liceum ogólnokształcącego z zakresu programowania mikrokontrolerów, jasno wskazują, iż jest to naturalne rozwinięcie programowania teoretycznego. Zbieżność języka programowania powoduje, że jedynym nowym elementem jest środowisko, w którym programujemy. Zastosowanie mikrokontrolerów Arduino przenosi uczniów w realny świat, gdzie mogą za pomocą rozwiązań informatycznych rozwiązywać realne problemy. Wymagana modyfikacja kodu programu, wymuszona dobrem odpowiednich przewodów, czy też zachowanie się poszczególnych podzespołów w silnych polach elektromagnetycznych, daje poczucie korelacji pomiędzy teorią a praktyką projektowanych rozwiązań. Możliwość taka jest nie do przecenienia, szczególnie w początkowym okresie nauki programowania, gdzie wyrobienie sobie pewnych nawyków tworząc kod z czasem doprowadzi do tworzenia bardziej stabilnych rozwiązań. Mając powyższe na uwadze, nauka programowania w różnych środowiskach, czyni z informatyki narzędzie praktyczne, pomocne w realizacji wielopłaszczyznowych zadań.

Omówione zadania realizowane na kółku informatycznym nie były badaniami naukowymi, ze względu na ograniczoną i niereprezentatywną grupę badanych. Stanowią jednak mogą odpowiedź i propozycję praktycznej realizacji zagadnień stawianych przez nową podstawę programową.

Na zakończenie pozostaje jedna, być może najważniejsza kwestia. Platforma Arduino lub jej podobne, jak np. Raspberry Pi, nie jest szeroko znana od strony praktycznej, większości nauczycieli informatyki w szkołach średnich. Pozostaje nadzieja, że wraz z wprowadzeniem nowej podstawy programowej pojawią się również adekwatne szkolenia dla nauczycieli w tym zakresie.

Appendix 1

```
// Czytnik numerów seryjnych czujników DS18B20
#include <OneWire.h>
// Numer pinu cyfrowego układu Arduino do którego podłączone są czujniki
const byte ONEWIRE_PIN = 2;
OneWire onewire(ONEWIRE_PIN);
void setup()
{
  while(!Serial);
  Serial.begin(9600);
}
void loop()
{
  byte address[8];
  onewire.reset_search();
  Serial.println("Paweł!");
  while(onewire.search(address))
  {
    if (address[0] != 0x28)
      continue;
    if (OneWire::crc8(address, 7) != address[7])
    {
      Serial.println(F("Błędny adres, sprawdź połączenia"));
      break;
    }
  }
  for (byte i=0; i<8; i++)
  {
    Serial.print(F("0x"));
    Serial.print(address[i], HEX);
    if (i < 7)
      Serial.print(F(", "));
  }
  Serial.println();
}
while(1);
}
```

Appendix 2

```
#include <Wire.h>
#include <OneWire.h>
#include <DS18B20.h>
#include <Servo.h>

// Numer pinu do którego podłączone są czujniki temperatury
#define ONEWIRE_PIN 2

Servo serwomechanizm1; //Tworzymy obiekt, dzięki któremu możemy odwołać się do serwa
int pozycja1 = 10; //Aktualna pozycja serwa dół 0-180

//Ustawienia bazowe !!!
float delta = 3; //różnica temperatur (włącz-wyłącz)
float Tstart = 22.70; //Temperatura otwarcia śluz powierza

int pozycja = 60; //Wstępna pozycja serwa
int zmiana = 4; //Co ile stopni ma się zmieniać pozycja serwa?

// Ilość czujników temperatury
#define SENSORS_NUM 2

// Adresy czujników temperatury odczytane uprzednio opisanym już programem
const byte address[SENSORS_NUM][8] PROGMEM = {
```

```

0x28, 0x3C, 0xEE, 0xC3, 0x16, 0x13, 0x1, 0x3B,
0x28, 0xFF, 0x12, 0xC3, 0x31, 0x18, 0x1, 0xB7
};

OneWire onewire(ONEWIRE_PIN);
DS18B20 sensors(&onewire);
//Port przekaźnika wiatraka nr 3
int SW = 3;

void setup()
{
  while(!Serial);
  Serial.begin(9600);

  pinMode(SW, OUTPUT);
  Wire.begin();
  sensors.begin();
  sensors.request();
  serwomechanizm1.attach(5); //Serwomechanizm1 (dół) podłączony do pinu 5
  //serwomechanizm1.writeMicroseconds(2000);
}

void loop() {
  if (sensors.available())
  {

    float temperature0 = sensors.readTemperature(FA(address[0]));
    float temperature1 = sensors.readTemperature(FA(address[1]));
    float roznica;

    Serial.print(F("#"));
    Serial.print("gora");
    Serial.print(F(": "));
    Serial.print(temperature0); //temperatura górna kolektora
    Serial.println(F(" C"));

    Serial.print(F("#"));
    Serial.print("dol");
    Serial.print(F(": "));
    Serial.print(temperature1); //temperatura dolna kolektora
    Serial.println(F(" C"));

    //logika
    roznica=temperature0-temperature1;

    if(temperature0>Tstart)
    {
      if(pozycja>38)
      {
        do
        {
          pozycja=pozycja-zmiana;
          serwomechanizm1.write(pozycja); //Wykonaj ruch serwa otwórz
          Serial.print(F("otwieranie: "));
          Serial.println(pozycja);
          delay(300);
        }while(pozycja>38);

        Serial.println("Śluza OTWARTA");
      }
    }
    if(Tstart>temperature0)
    {

```

```

    if(pozycja<85)
    {
    do
    {
    pozycja=pozycja+zmiانا;
    serwomechanizm1.write(pozycja); //Wykonaj ruch serwa zamknij
    Serial.print(F("zamykanie: "));
    Serial.println(pozycja);
    delay(300);
    }while(pozycja<85);
    delay(300);
    serwomechanizm1.write(85);
    }

    Serial.println("Śluza ZAMKNIĘTA");
    }
    if((roznica>delta)&&(temperature0>Tstart))
    {
    Serial.println("Wiatrak START");
    digitalWrite(SW,HIGH);
    // serwomechanizm1.write(100); //Wykonaj ruch serwa otwórz
    }
    if(roznica<delta)
    {
    Serial.println("Wiatrak STOP");
    digitalWrite(SW,LOW);
    // serwomechanizm1.write(10); //Wykonaj ruch serwa zamknij
    }
    sensors.request();
    delay(60000); //odświeżanie co 1 min
    }
}

```

Appendix 3

```

#define numReadings 30 //deklaracja zmiennych
#define A 5
#include "DHT.h"
#define DHT11_PIN 7
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
DHT dht;
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Ustawienie adresu wyświetlacza na 0x27
int czas [A];
int liczba [A];
int czujnik = A2;
int x=0;
unsigned long aktualnyCzas = 0;
unsigned long zapamietanyCzas = 0;
unsigned long roznicaCzasu = 0;
long long milis = 0;
int CPM = 0;
double prom = 0;
int pomoc = 1;
int menu=0;
double maxx=0;
double dawka=0;
int pod=1;

void setup() {
    dht.setup(DHT11_PIN);
    lcd.begin(16,2);
    void clear() ;
}

```

```

pinMode(6, INPUT_PULLUP); //przycisk lewy
pinMode(4, INPUT_PULLUP); //przycisk srodkowy
pinMode(3, INPUT_PULLUP); //przycisk prawy
pinMode(5, OUTPUT); //buzzer

lcd.backlight(); // zalaczenie podwietlenia
lcd.setCursor(0,0);
lcd.print("PrzemyslawKinasz");
delay(500);
lcd.setCursor(1,2);
lcd.print("Dozymetr PWSW");

delay(1000);

lcd.clear() ;
}
void loop() {
aktualnyCzas = millis();
if (analogRead(czujnik)>12) //jeśli na pinie analogowym nr 2 pojawi sie napiecie z transoptora po przejsci
czastki
{
milis=millis();
roznicaCzasu = aktualnyCzas - zapamietanyCzas;
zapamietanyCzas = aktualnyCzas;
x=x+1;
CPM=60000/roznicaCzasu; //ilosc zliczen na minute
dawka=x*0.006666; //przyjeta dawka promieniowania
prom=CPM* 0.006666; //aktualne promieniowanie
digitalWrite(5,HIGH); //buzzer off
if (maxx<prom) maxx=prom; //przypisywanie maksymalnej wartosci
delay(70); //czas na wygaszenie
digitalWrite(5,LOW); //buzzer off
}
delay(10);

if (menu==0) { //aktualny poziom promieniowania i liczba zliczen czastek

lcd.setCursor(0,0);
lcd.print("Prom: ");
lcd.setCursor(0,1);
lcd.print(prom);
lcd.setCursor(5,1);
lcd.print("uS/h");
lcd.setCursor(10,0);
lcd.print("Liczba: ");
lcd.setCursor(10,1);
lcd.print(x); delay(10);

if (digitalRead(4) == LOW) //reset
{
x=0;
dawka=0;
maxx=0;
prom=0;
delay(50);
lcd.clear() ;
}
}

if (menu==1) { //przyjeta dawka podczas sesji i wartosc maksymalna

lcd.setCursor(0,0);
lcd.print(" Dawka:");

```

```

        lcd.setCursor(0,1);

lcd.print(dawka);
        lcd.setCursor(5,1);
lcd.print("uS");
        lcd.setCursor(9,0);
        lcd.print("Max: ");
        lcd.setCursor(8,1);
lcd.print(maxx); delay(10);
        lcd.setCursor(12,1);
lcd.print("uS/h");
if (digitalRead(4) == LOW) //reset
{
    x=0;
    dawka=0;
    maxx=0;
    prom=0;
    delay(50);
    lcd.clear() ;
}
if (menu==2) { //temperatura i wilgotnosc

    lcd.setCursor(0,0);
        lcd.print("Temp: ");
        lcd.setCursor(0,1);
lcd.print(dht.getTemperature());
        lcd.setCursor(6,1);
lcd.print("C");
        lcd.setCursor(10,0);
        lcd.print("Wilg: ");

        lcd.setCursor(15,1);
lcd.print("%");
        lcd.setCursor(10,1);
lcd.print(dht.getHumidity()); delay(10);
}
if (menu==3) {

    lcd.setCursor(1,0);
        lcd.print("Podswietlenie: ");

    if (digitalRead(4) == LOW&&pod==1) //podswietlenie OFF
    {
        lcd.clear();
        lcd.noBacklight();
        pod=0;
        delay(200);
        lcd.setCursor(7,1);
        lcd.print("OFF");
    }
    if (digitalRead(4) == LOW&&pod==0) //podswietlenie ON
    {
        lcd.clear();
        lcd.backlight();
        pod=1;
        delay(200);
    }
}
if (digitalRead(3) == LOW&&menu<=2) { menu++; lcd.clear(); delay(200); } //sterowanie menu
if (digitalRead(6) == LOW&&menu>=1) { menu--; lcd.clear() ;delay(200); }
}

```

Bibliografia

- Anderson R., Cervo D., *Arduino dla zaawansowanych*, Helion, Gliwice 2014.
- Cathleen S., *Elektronika dla bystrzaków*, Septem, Gliwice 2017.
- Kisiel P., *Grafika komputerowa i informatyczne modelowanie struktur przestrzennych w programie kształcenia liceum ogólnokształcącego*, „Dydaktyka Informatyki” 2017.
- Kisiel P., *Praktyczne aspekty nauki programowania w szkole średniej*, „Dydaktyka Informatyki” 2018.
- Minnick C., Holland E., *Podstawy programowania dla młodych bystrzaków*, Septem, Gliwice 2016.
- Monk S., *Arduino dla początkujących. Podstawy i szkice*, Helion, Gliwice 2018.
- Monk S., *Elektronika z wykorzystaniem Arduino i Rapsberry Pi. Receptury*, Helion, Gliwice 2018.
- Platt C., *Elektronika od praktyki do teorii*, Helion, Gliwice 2013.

Netografia

- <http://www.arduino.cc/en/Main/Education>.
- http://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a.
- http://akademia.nettigo.pl/ds18b20/#co_to_jest_ds18b20.