

Data oddania: _____

Ocena: _____

KONRAD PŁAWIK	191458
Vladislav Mazur	199185
Łukasz Połubiński	211833

Podlewaczka*

* SVN: <https://github.com/KP191458/Wbudowane>

Spis treści

1. Opis projektu	3
1.1. Wykaz urządzeń	3
1.2. Wykaz funkcjonalności	3
2. Dokumentacja użytkownika	3
2.1. Wymagania sprzętowe	3
2.2. Instrukcja użytkownika	3
2.3. Interfejs użytkownika	4
3. Specyfikacja urządzeń	4
3.1. Arduino Uno	4
3.2. SHT31 - cyfrowy czujnik temperatury i wilgotności	5
3.3. Arduino-Dem - moduł wyświetlacza LCD 2"	6
3.4. Serwomechanizm Tower Pro SG90	6
3.5. Diody LED 5 mm	7
3.6. Rezystor przewlekany 330 Ω	7
3.7. Płytki stykowe 830 otworów	7
4. Specyfikacja funkcjonalności	7
4.1. SPI	7
4.2. GPIO	8
4.3. LCD	8
4.4. I2C	9
4.5. SHT31	10
4.6. Serwomechanizm	11
4.7. LED	11
Literatura	11

1. Opis projektu

Zadaniem projektu było stworzenie urządzenia które służyć będzie jako pomoc w monitorowaniu temperatury i wilgotności w ogrodzie bądź szklarni i dozowaniu zasobów wodnych. Urządzenie mierzyć będzie temperaturę otoczenia i wilgotność powietrza. Pomierzone dane są wyświetlane na ekranie a także urządzenie posiada dwie diody które sygnalizują przekroczenie ustalonego zakresu wilgotności. Przekroczenie zakresu uruchamia także serwomechanizm, który może służyć jako urządzenie otwierające zawór nawodnienia szklarni.

1.1. Wykaz urządzeń

- Arduino Uno
- SHT31 - cyfrowy czujnik temperatury i wilgotności
- Arduino-Dem - moduł wyświetlacza LCD 2"
- Serwomechanizm Tower Pro SG90
- Diody LED 5 mm
- Rezystory przewlekane 330 Ω
- Płytki stykowa 830 otworów
- Przewody połączeniowe męsko-męskie
- Przewody połączeniowe damsko-męskie

1.2. Wykaz funkcjonalności

1. SPI
2. GPIO
3. Timer
4. LCD
5. SHT31 - temperatura
6. SHT31 - wilgotność
7. Serwomechanizm
8. LED

2. Dokumentacja użytkownika

2.1. Wymagania sprzętowe

Projekt wymagał xxxx co wymagało modelu Arduino który posiadał co najmniej taką ilość wejść i wyjść. Zdecydowaliśmy się na użycie Arduino Uno, ponieważ posiadało ono wymaganą liczbę wejść i wyjść oraz, ze względu na popularność posiadało liczne dokumentacje i przykłady użycia.

2.2. Instrukcja użytkownika

Po podłączeniu zasilania zainstalowane na urządzeniu oprogramowanie rozpoczyna inicjalizację urządzeń a następnie pomiar temperatury i wilgotności.

- Pomiar wykonywany jest co 1 sekundę

- Odczytane pomiary wyświetlane są na wyświetlaczu LCD
- Dioda LED w kolorze zielonym informuje że zmieciona wilgotność jest poniżej 70%.
- Dioda LED w kolorze czerwonym informuje że zmieciona wilgotność jest powyżej 70%.
- Po przekroczeniu wilgotności 70% uruchomiony zostaje serwomechanizm który wykorzystany może zostać do otwarcia zaworu systemu nawadniającego (w szklarni bądź pomieszczeniu)

2.3. Interfejs użytkownika

Urządzenie nie wymaga interakcji użytkownika w celu dokonania pomiarów. Interfejsem jest ekran oraz kolor diod LED.

Z ekranu możemy odczytać wartości liczbowe temperatury (w stopniach Celsjusza) oraz wilgotności (w procentach).

3. Specyfikacja urządzeń

3.1. Arduino Uno

Arduino Uno to płyta mikrokontrolera oparta na ATmega328. Posiada 14 cyfrowych wejść / wyjść (z których 6 można wykorzystać jako wyjścia PWM), 6 wejść analogowych, ceramiczny rezonator 16 MHz, złącze USB, gniazdo zasilania, nagłówek ICSP i przycisk resetowania.

Arduino Uno w wersji R3 to najnowsza wersja po Duemilanove, z ulepszonym układem interfejsu USB. Podobnie jak Duemilanove, ma on nie tylko rozszerzony nagłówek osłony z napięciem odniesienia 3,3 V i pin RESET (który rozwiązuje problem dotarcia do szpilki RESET w osłonie) oraz bezpiecznik 500 mA do ochrony portu USB komputera, ale także automatyczny obwód do wyboru zasilania USB lub DC bez zworki! Uno jest kompatybilny z kodami pin i Duemilanove, Diecimilla i starszymi Arduino, więc wszystkie płytki (shields), biblioteki i kod będą nadal działać. Nowy R3 (trzecia wersja) UNO ma kilka drobnych aktualizacji, z aktualizacją układu interfejsu USB i dodatkowymi przełączaniami dla pinów i2c i pinów IOREF.

Arduino to platforma prototypowania elektroniki typu open source oparta na elastycznym, łatwym w użyciu sprzęcie i oprogramowaniu. Jest przeznaczony dla artystów, projektantów, hobbystów i wszystkich zainteresowanych tworzeniem interaktywnych obiektów lub środowisk.

Arduino może wyczuwać środowisko, otrzymując dane wejściowe z różnych czujników i może wpływać na otoczenie, kontrolując światła, silniki i inne siłowniki. Mikrokontroler na płycie programowany jest za pomocą języka programowania Arduino (opartego na Wiring) i środowiska programistycznego Arduino (opartego na Processing). Projekty Arduino mogą być autonomiczne lub mogą komunikować się z oprogramowaniem działającym

na komputerze (np. Flash, Processing, Max / MSP).

Parametry techniczne:

- Mikrokontroler ATmega328
- Napięcie robocze 5 V.
- Napięcie wejściowe (zalecane) 7-12 V.
- Napięcie wejściowe (limity) 6-20 V.
- Cyfrowe piny we / wy 14 (z których 6 zapewnia wyjście PWM)
- Piny wejścia analogowego 6
- Prąd DC na pin we / wy 40 mA
- Prąd DC dla 3,3 V Pin 50 mA
- Pamięć flash 32 KB (ATmega328) z czego 0,5 KB jest używane przez bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Szybkość zegara 16 MHz
- Długość 68,6 mm
- Szerokość 53,4 mm
- Waga 25 g

3.2. SHT31 - cyfrowy czujnik temperatury i wilgotności

Seria czujników temperatury i wilgotności SHT3x dostępna jest w dwóch wersjach: wersja niskokosztowa SHT30 i wersja standardowa sht31. Rodzina SHT3x łączy wiele funkcji i interfejsów z przyjaznym szerokim zakresem napięć roboczych (2.4 do 5.5 V). W porównaniu do czujników poprzedniej generacji seria SHT3x jest inteligentniejsza, bardziej niezawodna i dokładniejsza. Jest bardziej funkcjonalny, ma wyższe możliwości przetwarzania sygnału i może odczytywać wartości temperatury i wilgotności za pomocą różnych pinów.

Ponadto wprowadzenie tej wersji dodatkowo rozszerzyło rodzinę produktów SHT3x. Zastosowanie osłony ochronnej typu filmowego rozszerza również zakres zastosowań. Jest to stopień ochrony IP67 dla wody i ochrona przeciwpyłowa niż folia PTFE, dzięki czemu może być stosowany w trudnych warunkach, w których woda i kurz mogą wpływać na dokładność i wydajność czujnika.

Cechy:

1. Wysoka niezawodność i długa stabilność
2. Z technicznego punktu widzenia niezawodne
3. Nadaje się do aplikacji wysoka głośność
4. Wysoka zdolność przetwarzania sygnału
5. Niski poziom sygnału

Parametry produktu:

1. Wyjście: I2C, napięcie wyjściowe
2. Napięcie zasilania: 2.4 do 5.5 V

3. Zakres roboczy RH: 0-99.99% RH
4. Zakres temperatury pracy: -40 °C do 125 °C (-40 °F do 257 °F)
5. Czas reakcji RH: 8 sekund (τ_{63} %)

3.3. Arduino-Dem - moduł wyświetlacza LCD 2"

DEM 1280640 FGH-PW

FUNKCJE ORAZ CECHY

LCD TYPE Transflective Positive Mode

Viewing Direction : 6 o'clock

Driving Scheme : 1/65 Duty, 1/9 Bias

Power Supply Voltage : 3.3 Volt

VLCD : 9.0 Volt

Display Contents : 128 x 64 Dots

Driver IC : ST7565R (Sitronix)

RoHS : Compliant

Interfejs : SPI

SPECYFIKACJE MECHANICZNE

Module Size : 58.20 x 41.70 x 5.70 mm

Viewing Area : 50.00 x 25.00 mm

Active Area : 46.06 x 23.02 mm

Dot Size : 20.34. x 0.34 mm

Dot Gap : 20.02 mm

3.4. Serwomechanizm Tower Pro SG90

9-gramowe serwo typu micro SG90 firmy TowerPro to bardzo popularny model serwa stosowany w modelarstwie, robotyce, konstrukcjach napędowych, systemach alarmowych.

Serwo ma niewielką wagę lecz jest stosunkowo szybkie i ma siłę odpowiednią do sterowania modelami.

Charakterystyka:

- Prędkość przekładni: 0,12 s/60° (4,8 V)
- Moment: 1,2 – 1,8 kg/cm (4,8 V)
- Napięcie pracy: 4,8 – 7,2 V
- Długość przewodów zasilania: 23,5 cm
- Waga: 9 g
- Czas reakcji: 7 us
- Temperatura pracy: -30 do +60 stopni
- Wymiary 22 mm x 12 mm x 22,7 mm

Oznaczenie przewodów:

- Czerwony: zasilanie
- Brązowy: masa

- Pomarańczowy: sygnał

3.5. Diody LED 5 mm

Parametry diody zielonej:

- Soczewka w kolorze zielonym
- Obudowa: DIP 5 mm
- Długość emitowanej fali: 571 nm
- Jasność: 100 - 150 mcd
- Kąt świecenia: 50°
- Temp. pracy: od -40 °C do +80 °C
- Parametry pracy:
 - Prąd If: 20 mA
 - Napięcie Vf: 2,3 - 2,5 V

Parametry diody czerwonej:

- Soczewka w kolorze czerwonym
- Obudowa: DIP 5 mm
- Długość emitowanej fali: 625-645 nm
- Jasność: 450 - 800 mcd
- Kąt świecenia : 70 °
- Temp. pracy: od -40 °C do +80 °C
- Parametry pracy:
 - Prąd If: 20mA
 - Napięcie Vf: 2,0 - 2,3 V

3.6. Rezystor przewlekany 330 Ω

Specyfikacja rezystorów

- Rezystancja: 330 Ω
- Moc znamionowa: 1/4 W
- Tolerancja: 5%
- Montaż: przewlekany THT

3.7. Płytki stykowa 830 otworów

Dane techniczne płytki stykowej

- Wymiary: 165 x 53 mm
- Liczba otworów: 830
- Posiada kolorowe paski, które mogą oznaczać polaryzację zasilania (+i-)

4. Specyfikacja funkcjonalności

4.1. SPI

Interfejs SPI(Serial Peripheral Interface) używany jest do komunikacji z wyświetlaczem i diodami LED. Inicjalizacja obsługi wyświetlacza odbywa się poprzez załączenie biblioteki oraz wywołanie konstruktora identyfikującego urządzenie. W naszym wypadku jest to:

Przekazujemy do niego następujące wartości:

rotation: czyli kąt obrotu zawartości wyświetlacza (w naszym wypadku 180 stopni)

SCL (clock): numer pinu timera (sygnał taktujący) (PIN 13)

SDA (data): numer pinu przesyłu danych (PIN 11)

CS: wybór układu podrzędnego (PIN 10)

D/C (dc): numer pinu odpowiedzialny za wybór pomiędzy trybem instrukcji a trybem zanków (PIN 9)

reset: numer pinu odpowiedzialny za restart ekranu (PIN 8)

clock, data, cs, dc, reset mają wartości jednobajtowe.

4.2. GPIO

GPIO (General Purpose Input/Output) . GPIO kontroluje piny, ustawiając ich stan na wysoki (1 - HIGH) lub niski (0 - LOW) co, odpowiednio, skutkuje ustawieniem pina na urządzenia wyjścia jak i wejścia. W projekcie naszym użyliśmy GPIO do zmiany stanu dwóch diod LED sygnalizujących kolorem poziom wilgotności.

Fragmenty kodu podpowiadające za użycie GPIO:

```
int czerwona = 7;
int zielona = 6;

pinMode(czerwona , OUTPUT);
pinMode(zielona , OUTPUT);

if (h>70.0)
{
    digitalWrite(czerwona , HIGH);
    digitalWrite(zielona , LOW);
}
else
{
    digitalWrite(czerwona , LOW);
    digitalWrite(zielona , HIGH);
}
```

4.3. LCD

Użyta przez nas funkcjonalność niezbędna do podstawowej obsługi wyświetlacza przedstawiona została w rozdziale dotyczącym SPI. Natomiast jeśli zajdzie taka potrzeba można dokonać zaawansowanej obsługi wyświetlacza poprzez wpisanie porządkanych wartości w konkretne rejestry urządzenia.

Architekturę urządzenia przedstawia diagram:

Pełna lista rejestrów cmds znajduje się pod adresem:
https://github.com/olikraus/u8g2/blob/master/doc/controller_cmds.txt

Najważniejsze z nich to:

0x000	00000000	Lo Col
0x00f	00001111	Lo Col
0x010	00010000	Hi Col
0x01f	00011111	Hi Col
0x020	00100000	Res Ratio
0x023	00100011	Res Ratio
0x024	00100100	Res Ratio
0x027	00100111	Res Ratio
0x028	00101000	Pwr Ctrl
0x02f	00101111	Pwr Ctrl
0x040	01000000	Start Line
0x07f	01111111	Start Line
0x081	10000001	Contrast
0x0a0	10100000	Seg Normal
0x0a1	10100001	Seg Reverse
0x0a2	10100010	1/9 bias
0x0a3	10100011	1/7 bias
0x0a4	10100100	Normal Op
0x0a5	10100101	All pt on
0x0a6	10100110	Normal Op
0x0a7	10100111	Inverse
0x0ac	10101100	Indic. On
0x0ad	10101101	Indic. Off
0x0ae	10101110	Disp Off
0x0af	10101111	Disp On
0x0b0	10110000	Page Adr
0x0b7	10111111	Page Adr
0x0c0	11000xxx	Com Normal
0x0c7	11000xxx	Com Normal
0x0c8	11001xxx	Com Reverse
0x0cf	11001xxx	Com Reverse
0x0e0	11100000	Start R-M-W
0x0e2	11100010	Reset

4.4. I2C

Magistrala I2C zawiera siedem rejestrów:

- **I2CONSET** (Control Set register) Odpowiada za ustawianie bitów w rejestrze I2CON, który kontroluje operacje na interfejsie I 2 C. Ustawienie jedyńki w tym rejestrze skutkuje ustawieniem odpowiedniego bitu w

- rejestrze kontrolnym I²C. Ustawienie stanu niskiego (zera) nie powoduje żadnych efektów. I2CONSET zawiera osiem bitów.
- **I2STAT** (Status register) Odpowiada za informacje o I²C, rejestr w trybie tylko do odczytu. Również posiada osiem bitów, z czego trzy pierwsze (bity o numerze 0,1,2) są nieużywane i są zerami. Pozostałe bity o numerach 3,4,5,6,7 dostarczają informacji statusie interfejsu I²C.
 - **I2DAT** (Data register) Rejestr ten zawiera wysłane lub otrzymane dane. Proces może odczytać i zapisywać ten rejestr tylko wtedy, gdy nie jest w trakcie przesuwania bitu
 - **I2ADR** (Slave Address register) Rejestr ten jest możliwy do odczytu i zapisu. Używany tylko wtedy, gdy interfejs I²C jest ustawiony na slave mode. W innym przypadku rejestr nie jest używany.
 - **I2SCLH** (High duty cycle register) Rejestr ten zawiera 16 bitów (od 0 do 15). Ustawia czas SCL HIGH zegara I²C.
 - **I2SCLL** (Low duty cycle register) Rejestr ten zawiera 16 bitów (od 0 do 15). Ustawia czas SCL LOW zegara I²C.
 - **I2CONCLR** (Control Clear register) Odpowiada za czyszczenie bitów w rejestrze I2CON. Ustawienie jedynki w tym rejestrze skutkuje ustawieniem odpowiedniego bitu w rejestrze kontrolnym I²C. Ustawienie zera nie powoduje żadnych efektów. Rejestr ten posiada osiem bitów.

W projekcie naszym używaliśmy jedynie rejestru **I2ADR**.

4.5. SHT31

Odczyt z termometru/miernika wilgotności SHT31 opiera się o odczyt danych z wymienionego wyżej rejestru I2ADR. Do zaawansowanej obsługi SHT31 możemy użyć jego rejestrów:

nazwa	wartość	opis
SHT31_DEFAULT_ADDR	0x44	Adres domyślny
SHT31_MEAS_HIGHREP_STRETCH	0x2C06	Measurement High Repeatability with Clock Stretch Enabled
SHT31_MEAS_MEDREP_STRETCH	0x2C0D	Measurement Medium Repeatability with Clock Stretch Enabled
SHT31_MEAS_LOWREP_STRETCH	0x2C10	Measurement Low Repeatability with Clock Stretch Enabled
SHT31_MEAS_HIGHREP	0x2400	Measurement High Repeatability with Clock Stretch Disabled
SHT31_MEAS_MEDREP	0x240B	Measurement Medium Repeatability with Clock Stretch Disabled
SHT31_MEAS_LOWREP	0x2416	Measurement Low Repeatability with Clock Stretch Disabled
SHT31_READSTATUS	0xF32D	Read Out of Status Register
SHT31_CLEARSTATUS	0x3041	Clear Status
SHT31_SOFTRESET	0x30A2	Soft Reset
SHT31_HEATEREN	0x306D	Heater Enable
SHT31_HEATERDIS	0x3066	Heater Disable
SHT31_REG_HEATER_BIT	0x0d	Status Register Heater Bit

4.6. Serwomechanizm

Serwomechanizm wykorzystuje interfejs SPI. Obsługa serwomechanizmu odbywa się za pomocą biblioteki Servo. Przy inicjalizacji przypisujemy numer pinu a następnie przy pomocy funkcji write podajemy kąt jaki wskazać ma serwomechanizm.

4.7. LED

Literatura

Na końcu należy obowiązkowo podać cytowaną w sprawozdaniu literaturę, z której grupa korzystała w trakcie prac nad zadaniem (przykład na końcu szablonu)