



Instytut Informatyki Politechniki Śląskiej  
Zespół Mikroinformatyki i Teorii Automatów  
Cyfrowych

**Rok akademicki:**

Rodzaj studiów\*: SSI/NSI/NSM

**Przedmiot (Języki  
Asemblerowe/SMiW):**

Grupa

## Sekcja

# 2017/2018

NSI

**SMiW****BDIS**

3

**Imię:**

# Bartłomiej

**Prowadzacy:**

OA/JP/KT/GD/BSz/GB

**GB**

**Nazwisko:**

# Skrobol

## Raport końcowy

**Temat projektu:**

## Automatyczne nawadnianie i kontrola parametrów klimatycznych w mini szklarni

**Data oddania:**  
**08/08/2018**

# **Automatyczne nawadnianie i kontrola parametrów klimatycznych w mini szklarni**

## **Główne założenia projektu:**

1. Kontrola wilgotności gleby
2. Automatyczne nawadnianie
3. Kontrola poziomu wody w zbiorniku
4. Pomiary wilgotności powietrza, temperatury
5. Podgląd parametrów klimatycznych na wyświetlaczu

Najważniejszym celem projektu jest kontrola wilgotności gleby i w razie konieczności uruchomienie pompy wody w celu nawodnienia. Wilgotność mierzona jest za pomocą analogowego czujnika.

Poziom wody nadzorowany jest przy pomocy ultradźwiękowego czujnika odległości. W przypadku jej zbyt niskiego poziomu pompa się nie uruchomi.

Dodatkowo układ wyposażony jest w termometr i czujnik wilgotności powietrza. To wszystko mieści się w module z interfejsem 1-wire

Wszystkie parametry można podejrzeć na wyświetlaczu OLED pracującym na interfejsie SPI.

Całość pracuje na mikrokontrolerze Atmega32a. Głównymi cechami przemawiającymi za wyborem jest odpowiednia ilość SRAM, przetwornik analogowo cyfrowy, interfejs SPI oraz timer ICP (input capture) przy zachowaniu niskiej ceny.

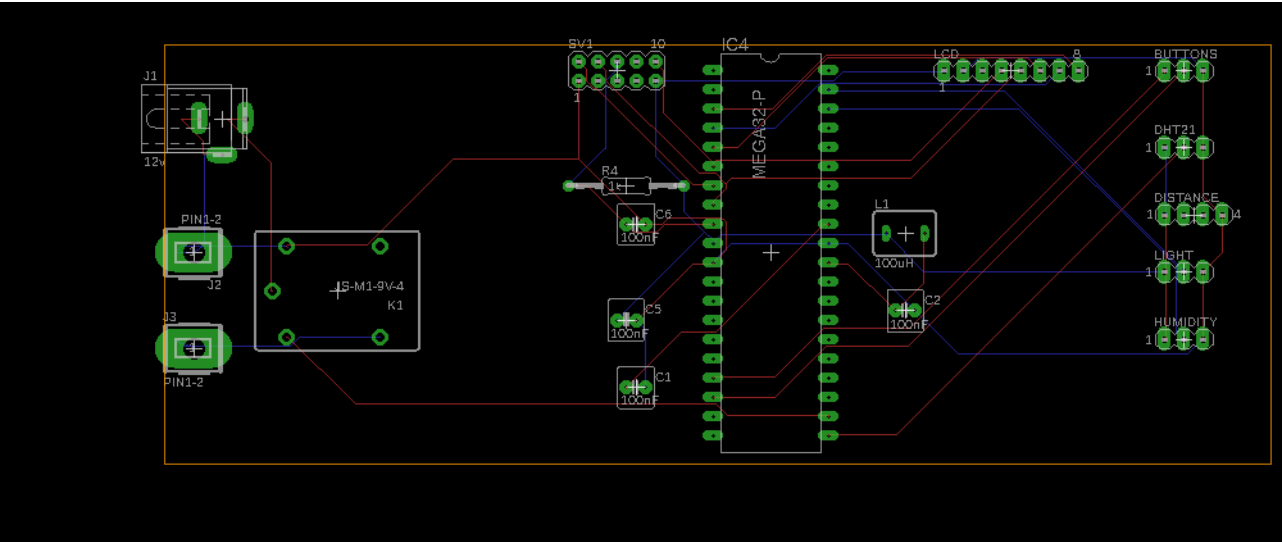
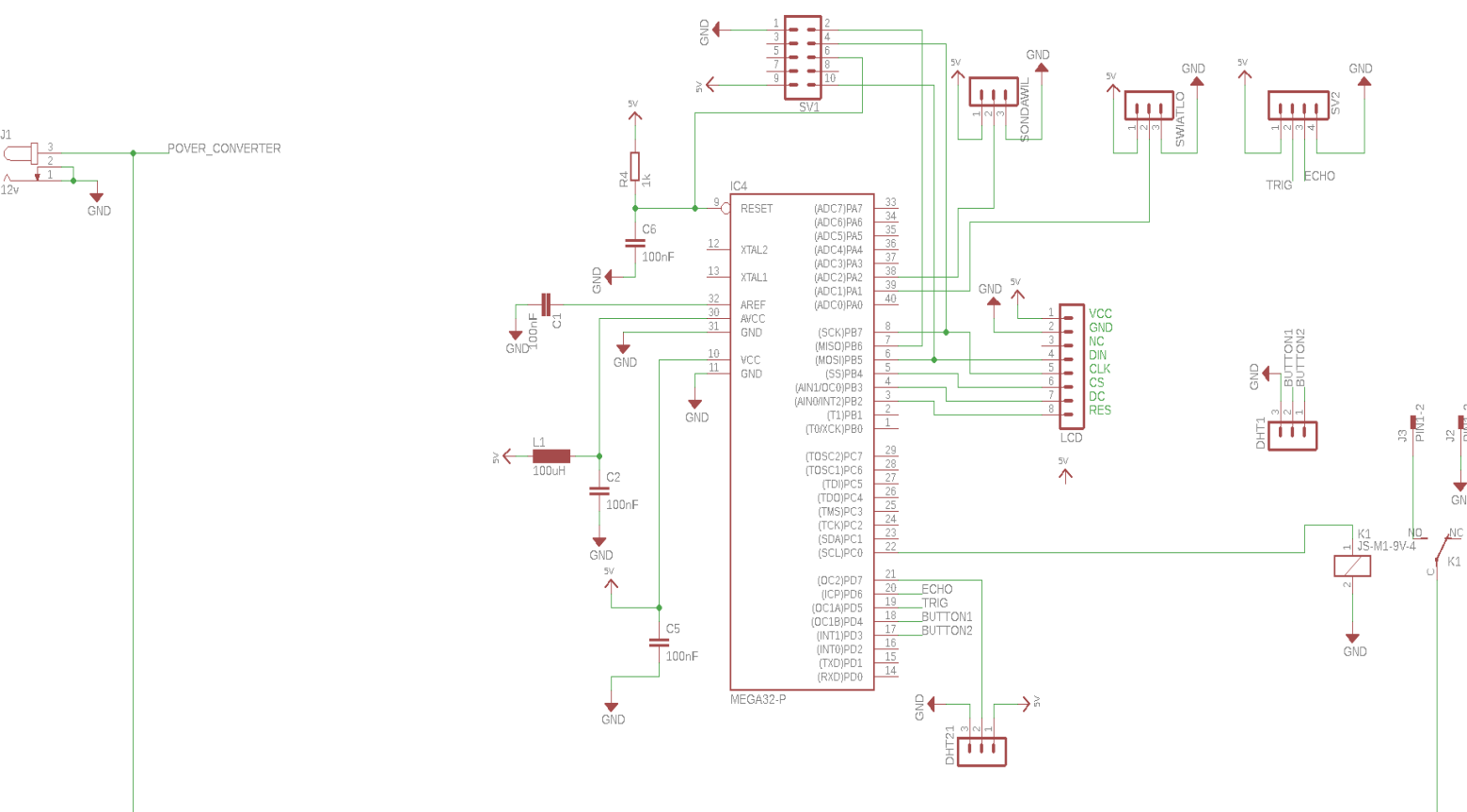
Większość podzespołów pracuje na zasilaniu 5V. Jedynie pompa pracuje na 12V. Z tego powodu użyłem zasilacza 12V a w układzie znajduje się przetwornica step-down obniżająca napięcie dla reszty elementów.

Do uruchomienia pompy potrzebny jest przekaźnik. Z powodu braku dostępności przekaźników o ułożeniu pinów pasującym do płytki uniwersalnej musiałem zakupić moduł z przekaźnikiem.

Całość zbudowana jest na płytkach uniwersalnych.

Za doбором elementów przemawiała cena oraz łatwość obsługi.

# Specyfikacja wewnętrzna



## Opis funkcji poszczególnych bloków układu

### 1. Czujnik wilgotności gleby :

Posiada wyjście analogowe podłączone do przetwornika analogowo-cyfrowego mikrokontrolera. Wartość napięcia zmniejsza się wraz ze wzrostem wilgotności.

### 2. Ultradźwiękowy czujnik odległości:

Po nadaniu stanu wysokiego na pin TRIG przez 10 us moduł dokonuje pomiaru. Do mikrokontrolera wysyłany jest wtedy sygnał którego czas trwania jest proporcjonalny do odległości.

Czas trwania sygnału mierzony jest za pomocą timera ICP

### 3. Wyświetlacz OLED:

Komunikacja z wyświetlaczem wykorzystuje protokół SPI

### 4. Termometr i czujnik wilgotności powietrza:

Powyższe elementy znajdują się w jednym module, z którym mikrokontroler komunikuje się za pomocą 1-wire.

### 5. Pompa wody:

Pompą steruje moduł z przekaźnikiem, z cyfrowym wejściem.

Nadanie stanu wysokiego przełącza styki przekaźnika uruchamiając pompę

### 6. Przetwornica step-down:

Obniża napięcie z 12V na 5V. Zapewnia większą wartość prądu niż stabilizator, przy zachowaniu niższej temperatury.

### 7. Klawiatura:

Dwa tact-switches z jednej strony podłączone do masy, z drugiej do mikrokontrolera. Pull-up wykonany jest programowo.

Po wciśnięciu przycisku na pin podawany jest potencjał masy.

## Lista wykorzystanych elementów:

- Mikrokontroler AVR - ATmega32A-PU - DIP
- Czujnik wilgotności gleby (bez nazwy)
- Ultradźwiękowy czujnik odległości HC-SR04
- Czujnik temperatury i wilgotności DHT21 (AM2301) w obudowie
- Pompa do cieczy 12V 110l/h - 7mm
- Dławik osiowy przeciwzakłócenia 100uH/90mA
- Zasilacz impulsowy 12V / 1,25A - wtyk DC 5,5 / 2,5mm
- Wyświetlacz OLED niebieski graficzny 1,3" (B) 128x64px SPI/I2C
- Moduł z przetwornicą impulsową step-down LM2596 3,2V-35V 3A
- Moduł przekaźnika 1 kanał - cewka 5V JQC-3FF-S-Z
- Płytki uniwersalne średnia 420 pól - MSX
- Gniazdo DC 2,5/5,5
- Złącza raster 2,54mm
- Tact Switch 6x6mm / 13mm THT
- Kondensatory ceramiczne 100nF/50V THT
- Rezystory THT
- Wtyk IDC 10

# Algorytm oprogramowania urządzenia

Inicjalizacja podzespołów

WHILE 1

    pobierz wilgotność gleby  
    pobierz poziom wody

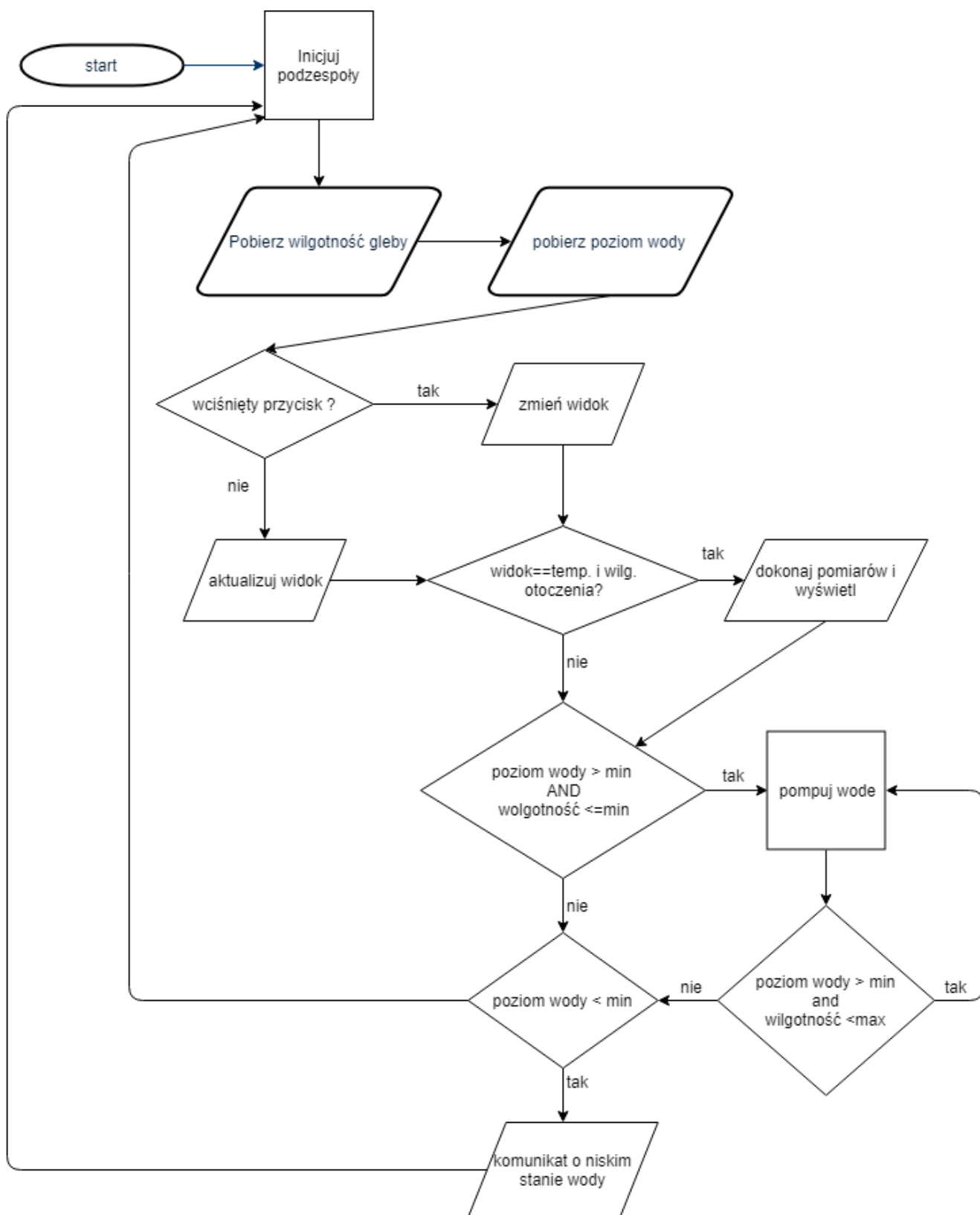
    IF wciśnięty przycisk  
        THEN zmień widok  
            IF widok = widok temperatura i wilgotność powietrza  
                THEN dokonaj pomiarów i wyświetl

    ELSE IF widok = widok temperatura i wilgotność powietrza  
        THEN dokonaj pomiarów i wyświetl

    ELSE aktualizuj widok

    IF poziom wody > min AND wilgotnosc <= min  
        THEN WHILE poziom wody > min AND wilgotnosc < max  
            pompa uruchomiona

    IF poziom wody < min  
        THEN wyświetl komunikat



## Opis klas

ADCReader - Umożliwia dokonywać pomiarów za pomocą 10 bitowego konwertera analogowo cyfrowego. Zakres napięcia jakie jest w stanie mierzyć to 0-5 V.

- `uint16_t read(uint8_t channel);` - dokonuje pomiaru kanale od 0 do 8. maksymalna wartość jaką może zwrócić funkcja to 1023. Wartość ta oznacza napięcie 5V.

HumiditySensor - Za pomocą tej klasy możemy zmierzyć wilgotność gleby analogowym czujnikiem

- `HumiditySensor(uint8_t channel);` - Konstruktor jako parametr przyjmuje pin do którego podłączony jest czujnik
- `volatile uint8_t measure( );` - dokonuje pomiaru, wartość zwracana wyrażona jest w procentach

ICPTimer - 16 bitowy timer mierzący czas trwania sygnały na pinie ICP1 (PD6), wyzwalany stanem wysokim. Wymaga by taktowanie procesora wynosiło 8Mhz

- `static uint32_t getTime();` - zwraca czas ostatniego pomiaru w mikrosekundach.

DistanceSensor - sterownik ultradźwiękowego czujnika odległości

- `float getDistance();` - zwraca dystans pomiędzy czujnikiem a najbliższą przeszkodą. Skuteczność to 2 - 200 cm. Wartość wyrażona w cm.

Button - klasa reprezentująca przycisk, reagujący na pojawienie się stanu niskiego. Drgania styków wyeliminowane są programowo

- `Button(uint8_t button);` - Konstruktor inicjujący przycisk, uruchamia wewnętrzny rezystor podciągający.
- `volatile bool isPressed();` - zwraca true jeśli przycisk jest wciśnięty.



- poziom wody większy niż wymagane minimum
- wilgotność gleby niższa niż minimalna wartość
- pompa uruchomiona jest tak długo puki wilgotność nie osiągnie maksimum

- `bool controll(volatile float & waterDistance, volatile uint8_t & humidity);` - `waterDistance` – pomiar poziomu wody  
`humidity` – pomiar wilgotności  
 Na podstawie tych wartości kontroler decyduje czy pompa ma być uruchomiona czy też nie. Zwraca `false` jeśli poziom wody jest zbyt niski,  
 w innym wypadku `true`.

- `uint8_t screen[8*128];` - publiczna zmienna buffora
- `void clear();` - czyści buffor
- `void show();` - wyświetla buffor
- `void string(uint8_t x, uint8_t y, const char *pString, uint8_t Size, uint8_t Mode, uint8_t* buffer);` -
  - x - początek rysowania w osi x
  - y - początek rysowania w osi y
  - \*pString - null terminated string który chcemy wyświetlić
  - Size - rozmiar czcionki
  - Mode - tryb wyświetlania 1: jasne litery , 0:ciemne litery na jasnym tle
  - buffer - buffor do którego chcemy pisać
- `static void restart();`

## Opis funkcji

spi.h – funkcje odpowiedzialne za interfejs SPI

- void spiInitMaster(void) – inicjalizacja SPI w trybie master
- void spiWrite(const uint8\_t& data) – wysyła pojedynczy bajt

dht.h – open source biblioteka do obsługi czujnika DHT21 czyli modułu z termometrem i czujnikiem temperatury

- int8\_t dht\_gettemperaturehumidity(int8\_t \*temperature, int8\_t \*humidity); - mierzy temperature i wilgotność, wyniki zapisuje w temperature i humidity. Zwraca -1 jeśli pomiar jest nieudany

ScreenController.h – zbiór funkcji wyświetlających odpowiednie widoki

- extern SH1106 screen; - globalny obiekt wyświetlacza
- void showHumidityScreen(volatile uint8\_t &hum); - zmienia widok na widok wilgotności gleby i wyświetla jej wartość podaną w hum  
void updateHumidityScreen(volatile uint8\_t &hum); - aktualizuje wyświetlaną wartość
- void showTempAndHumidityScreen(); - zmienia widok na widok temperatury i wilgotności powietrza, pobiera wartości z czujnika i wyświetla je
- void updateTempAndHumidityScreen(); - aktualizuje i wyświetla pomiary
- void showTankScreen(volatile float &distance); - zmienia widok na widok poziomu wody i wyświetla jego wartość podaną w distance
- void updateTankScreen(volatile float &distance); - aktualizuje wyświetlaną wartość

## Szczegółowy opis działania ważniejszych procedur

ADCReader::ADCReader(): inicjalizacja konwertera analogowo cyfrowego w trybie single conversion.

Do skonfigurowania konwertera służą 2 rejestry

1. ADMUX – pozwala na wybranie napięcia odniesienia, w tym przypadku jest to AVCC jak również wybranie kanału na którym ma być dokonany pomiar ale ta opcja ustawiana jest podczas pomiaru.
2. ADCSRA – ustawia preskaler oraz uruchamia konwerter. Przetwornik musi pracować z częstotliwością pomiędzy 50 – 200 kHz, taktowanie mikrokontrolera ustawione jest na 8MHz dlatego też preskaler jest ustawiony na 128.  $8000000/128 = 62500$  czyli 62kHz

ADCReader::read(uint8\_t): po zapewnieniu że kanał jest z zakresu 0 – 7 ustawiając bit ADSC w rejestrze ADSCRA rozpoczynamy pomiar na zadanym kanale. O skończonym pomiarze informuje nas zmiana stanu w rejestrze ADSCRA na bicie ADIF, zmienia się na wysoki, lub bit ADSC zmienia się na 0.

ICPTimer::ICPTimer(): konstruktor inicjalizuje 16 bitowy timer liczący długość sygnału na pinie ICP. W rejestrze TCCR1B ustawiamy jakim sygnałem aktywujemy licznik. W naszym przypadku jest to stan wysoki. Ustalamy również preskaler na 8, co daje nam 1 jednostkę czasu równą 1 mikrosekundzie.

W rejestrze TIMSK włączamy przerwania z przepełnienia bufora oraz zmiany stanu na wejściu ICP.

Mając włączone przerwania musimy jeszcze zaimplementować funkcje obsługujące je. Funkcja przerwania przepełnienia zwiększa licznik przepełnień. Funkcja przerwania spowodowanego zmianą stanu jeśli była ustawiona na reagowanie na zbocze wznoszące to zeruje licznik przepełnień, pobiera aktualny stan licznika oraz ustawia przerwanie w oczekiwaniu na zbocze opadające. Gdy pojawi się zbocze opadające nasz pomiar dobiegł końca, pobieramy stan licznika, wyliczamy długość trwania sygnału wysokiego z pomocą licznika przepełnień. Aktywator przerwań ustawiony na zbocze wznoszące.

# Specyfikacja zewnętrzna urządzenia

## Opis funkcji elementów sterujących urządzeniem

Tact switche – dokładniej to dwa, odpowiadają za przełączanie pomiędzy widokami

## Opis funkcji elementów wykonawczych

Wyświetlacz SH1106 – Informuje o aktualnym poziomie wody w zbiorniku, wilgotności gleby i powietrza oraz temperaturze.

Przełącznik JQC-3FF-S-Z – uruchamia pompę która pracuje na wyższym napięciu niż reszta układu.

Pompa wody – służy do podlewania rośliny

## Instrukcja obsługi

Należy umieścić czujnik wilgotności gleby w doniczce.

Nalać wody do zbiornika, oraz upewnić się że krótki wężyk znajduje się w zbiorniku, a długi w doniczce.

Wsadzić wtyczkę do kontaktu.

Klawisze służą do przełączania się pomiędzy wyświetlanymi parametrami. W przypadku zbyt niskiego poziomu wody zostanie wyświetlony komunikat u dołu ekranu.

## Testowanie

Początkowo część układu zbudowana była na płytce prototypowej. Jako pierwszy uruchomiłem wyświetlacz co pozwoliło mi w późniejszych etapach wyświetlać informacje niezbędne do testowania i debugowania układu. Po oprogramowaniu każdego kolejnego elementu sprawdzałem poprawność działania klasy/funkcji. Dopiero mając zaprogramowane wszystkie elementy zacząłem łączyć wszystko w całość.

## Wnioski

### Problemy które wystąpiły podczas montażu i uruchamiania

Dokonałem złego zakupu złączy i wtyków przez co podpinanie okablowania jest mało intuicyjne.

Po wyłączeniu się pompy wyświetlacz się zawieszał. Poradziłem sobie z tym programowo, resetuje go za każdym razem gdy wyłączam pompę.

Brak dokumentacji czujnika wilgotności spowodował jego uszkodzenie. Doszło do korozji sondy, spowodowanej ciągłym przepływem prądu. Najlepszym rozwiązaniem jest dostarczać napięcie przed samym pomiarem i wyłączać po zakończeniu pomiaru.

Podczas przylutowywania okablowania do czujnika światła spaliłem go.

### Testy poprawności działania urządzenia

Umieściłem czujnik wilgotności w wodzie. Po wyjęciu go z niej pompa się uruchamiała. Po ponownym włożeniu wyłączała się tak jak zakładałem. Drugim testem było oddalanie czujnika odległości, podczas pompowania by mierzył wartość większą od minimalnej co skutkowało wyłączeniem się pompy oraz wyświetleniem odpowiedniego komunikatu.

Cały projekt dostępny jest pod adresem <https://github.com/Skrobson/SMIW>

## Literatura

<http://mikrokontrolery.blogspot.com/>