

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji. Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.



**Uwaga! W urządzeniu występuje napięcie niebezpieczne dla życia! Należy zachować ostrożność podczas uruchamiania układu. Gałka enkodera powinna być wykonana z materiału, który nie przewodzi prądu.**

# Sterownik wentylatora wyciągu łazienkowego (kuchennego)

Ręczne sterowanie wentylatorem wyciągu w łazience czy kuchni nie jest zbyt wygodne. W łazience czy toalecie można podłączyć wentylator pod oświetlenie, jednak to rozwiązanie ma kilka wad. Wentylator wyłączy się natychmiast gdy wyjdziemy z łazienki po kąpieli, gdy w łazience jest jeszcze duża wilgotność. W kuchni nie ma innego rozsądnego rozwiązania niż sterowanie ręczne. Zaprezentowany sterownik automatycznie steruje wentylatorem na podstawie zmian wilgotności, czasowo po włączeniu/wyłączeniu oświetlenia oraz umożliwia sterowanie ręczne. Złącze karty sieciowej Wi-Fi umożliwia podłączenie sterownika do systemu inteligentnego domu.

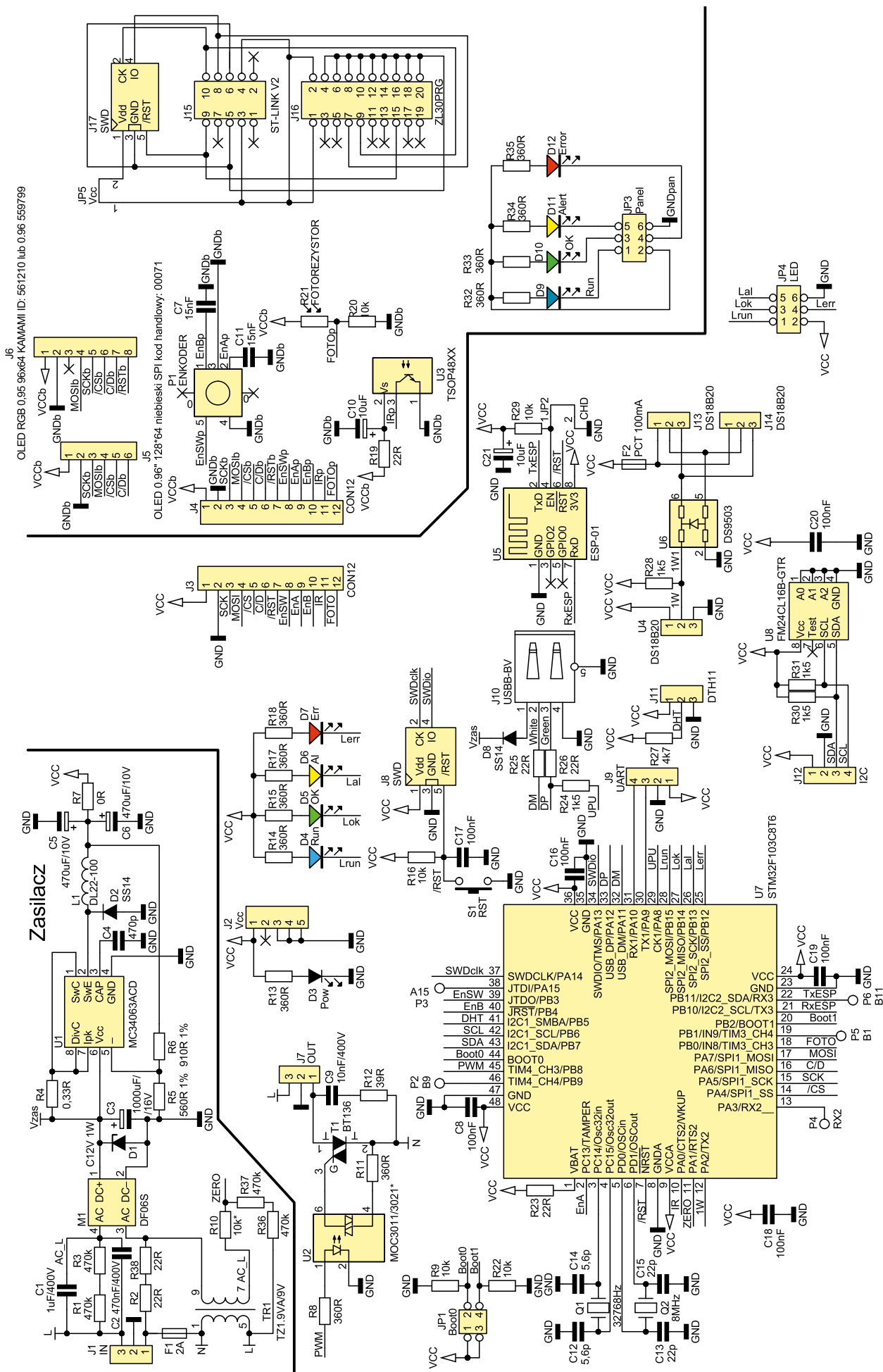
## Budowa i działanie

Schemat układu prezentuje **rysunek 1**. Urządzenie zasilane jest stabilizatorem impulsowym zbudowanym na układzie

U1 (MC34063ACD) pracującym w typowej aplikacji. Stabilizator może być zasilany z transformatora lub z zasilacza beztransformatorowego. Zasilacz beztransformatorowy

sprawdzi się w sytuacji, gdy nie korzystamy z łącza USB i pobór prądu jest niewielki (brak modułu Wi-Fi). Stabilizowane napięcie 3,3 V zasilają wszystkie obwody sterownika.

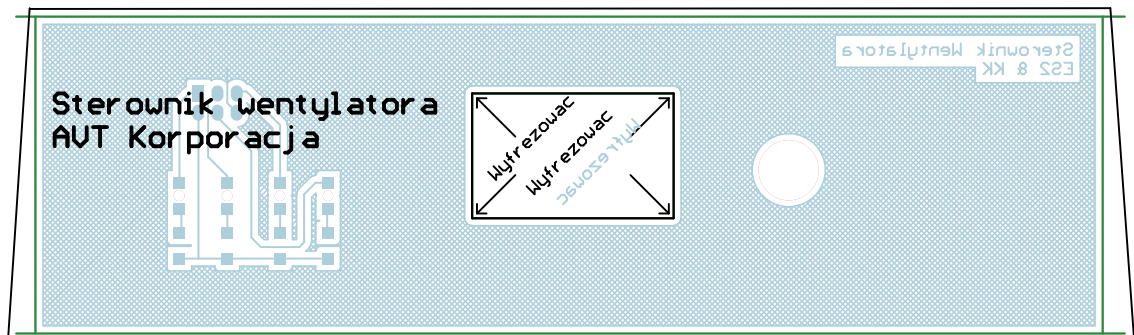
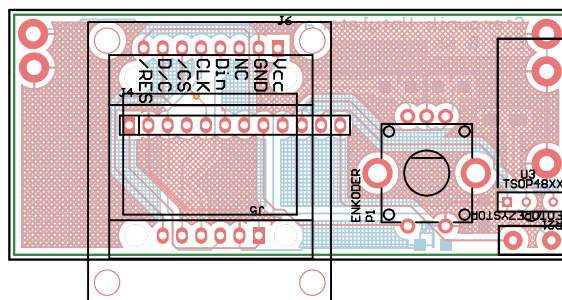
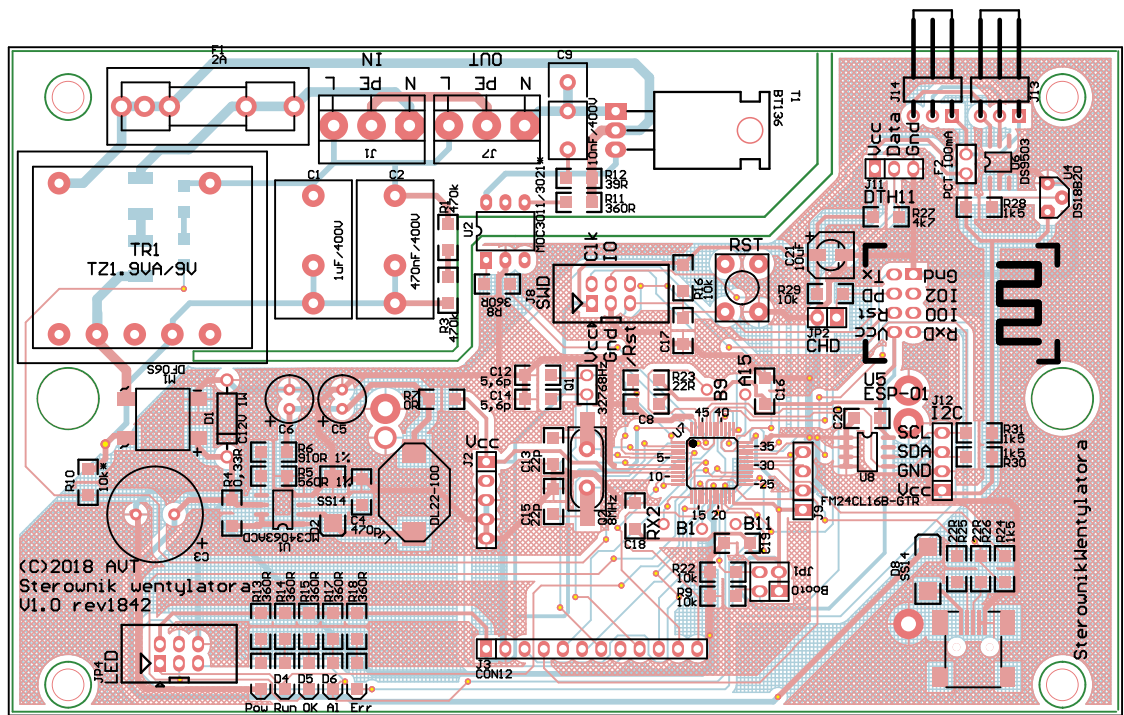
Mikrokontrolerem sterującym jest ARM STM32F103. Na płytce przewidziano kwarce do taktowania  $\mu C$  oraz zegara RTC. W aktualnej wersji oprogramowania  $\mu C$  jest taktowany wewnętrznym generatorem RC i kwarce nie są używane ale wyklucza to możliwość uruchomienia USB (wymagany kwarce). Elementem sterującym fazowo silnikiem wentylatora jest triak sterowany optotriakiem. Triak jest zabezpieczony szybkim bezpiecznikiem F1. Przejście napięcia sieciowego



Rysunek 1. Schemat układu

przez zero, wymagane przy sterowaniu fazowym, jest wykrywane przez GPIO PA2 (przerwanie EXT11). W przypadku zasilacza beztransformatorowego montowane są rezystory R36, R37, w przypadku zasilacza z transformatorem – montowany jest R10. Sterownik wykrywa zarówno narastanie jak i opadanie napięcia sieciowego (reakcja na oba zbrocza sygnału). Diody D4... D7 sygnalizują stan pracy sterownika.

Do komunikacji z użytkownikiem zastosowano enkoder i monochromatyczny, dwukolorowy, wyświetlacz OLED o rozdzielczości 128×64. Odbiornik podczerwieni (U3) umożliwia zdalne sterowanie urządzeniem ale ta opcja jest przewidziana do przyszłych zastosowań i nie została jeszcze oprogramowana. Wilgotność powietrza mierzy czujnik DTH11 podłączony do złącza J11. Istnieje możliwość pomiaru temperatury termometrami DS18B20 podłączonymi do J13, J14 i/ lub U4 wlotowanego w płytkę sterownika. U6 (DS9503) zabezpiecza wejście µC przed uszkodzeniem przepięciami, które mogą pojawić się na przewodach łączących termometry ze sterownikiem. Zasilanie czujników zewnętrznych jest zabezpieczone bezpiecznikiem polimerowym F2. Włączenie oświetlenia wykrywa fotorezystor R21 tworzący wraz z R20 dzielnik napięcia. Umożliwia to pomiar intensywności oświetlenia bo wyłączenie oświetlenia np. w łazience nie musi oznaczać, że jest całkowicie ciemno, przykładowo światło może wpadać przez szybę w drzwiach. Sterownik wyposażono w pamięć FRAM (U8) ale nie



Rysunek 2. Schematy płytek wraz z rozmieszczeniem elementów oraz rysunek płyty czołowej

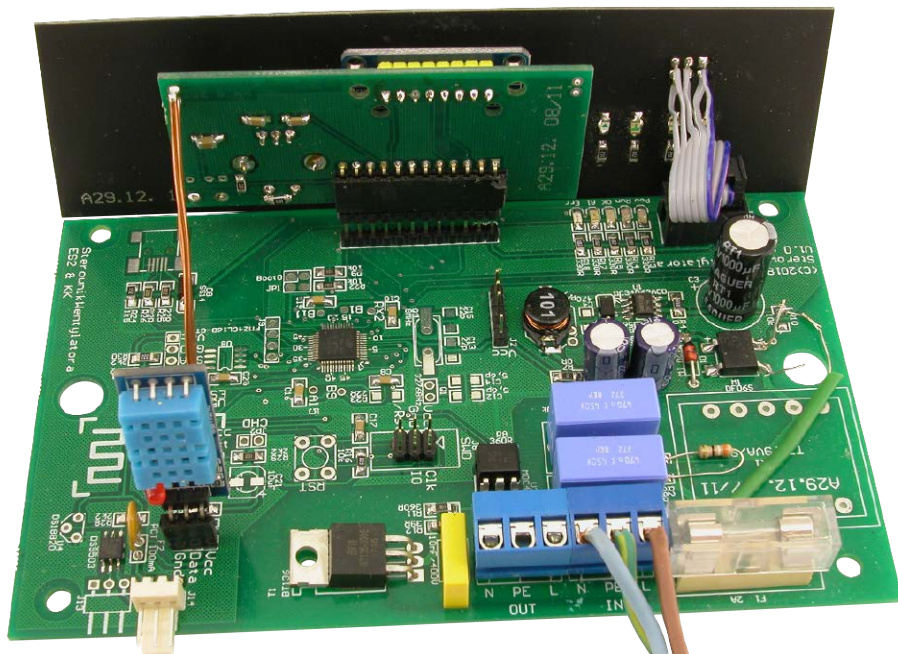
jest ona używana, ustawienia zapisywane są w EEPROM emulowanym w pamięci FLASH µC.

### Montaż i uruchomienie

Schemat płytki wraz z rozmieszczeniem elementów pokazano na **rysunku 2**. Na początku montażu należy wybrać sposób zasilania: transformatorem czy beztransformatorowo. Zależnie od opcji nie montuje się transformatora lub elementów C1, C2, R1... R3. Następnie należy zamontować elementy stabilizatora impulsowego i uruchomić go.

Gdy pracuje poprawnie można zamontować pozostałe elementy. Nie ma potrzeby montowania kwarców i kondensatorów do nich, odbiornika podczerwieni i pamięci FRAM.

Złącze J8 służy do zaprogramowania µC, do tego celu wystarczy ST-Link V2. Jeżeli w sterowniku zastosowano zasilacz beztransformatorowy, podczas programowania, należy zastosować izolowany galwanicznie interfejs USB, transformator separacyjny 230 V/230 V lub zasilic sterownik z izolowanego galwanicznie zasilacza 7...12 V. Jeśli czujnik wilgotności będzie zamontowany



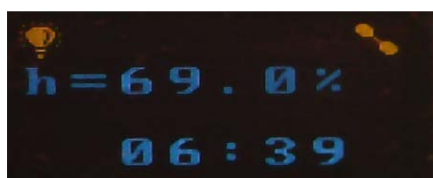
Fotografia 3. Widok zmontowanego układu

wewnątrz sterownika należy zadbać o odpowiednie otwory w obudowie. Ze względu na to, że elementy sterownika mogą się nagrzewać (transformator, triak) zaleca się zamontowanie czujnika poza sterownikiem.

Płyta czołowa sterownika jest wlotowana w płytę bazową z wykorzystaniem kąтового gniazda GOLDPIN oraz listwy o wysokości 20 mm. Aby ustabilizować konstrukcję, płyta czołowa jest wzmocniona dwoma drutami wlotowanymi w przygotowane do tego celu pola na płytkach (fotografia 3). Dzięki temu płyta czołowa nie ugina się podczas naciskania przycisku enkodera. Diody LED na płycie czołowej połączone są z płytą bazową z użyciem gniazda i wtyku IDC6FT. Jeżeli



Fotografia 4. Ekran powitalny



Fotografia 5. Wygląd ekranu głównego



Fotografia 6. Ekran konfiguracji poziomów wilgotności

zamontowane zostaną LED zarówno na płycie czołowej jak i na płycie głównej to należy zwiększyć wartość rezystorów ograniczających prąd do 750 Ω, zarówno na płycie czołowej jak i na głównej. Wymiary płytek i rozmieszczenie otworów są przystosowane do obudowy KM-50.

## Obsługa urządzenia

Jeśli montaż przeprowadzono prawidłowo, po włączeniu sterownika pokaże się ekran powitalny (fotografia 4), a po chwili ekran główny (fotografia 5). Przy pierwszym uruchomieniu sterownik należy skonfigurować. W tym celu kręcąc enkoderem wybieramy ekran konfiguracji poziomów wilgotności, przy których nastąpi załączenie i wyłączenie wentylatora (fotografia 6). W drugim wierszu wyświetlana jest aktualna wilgotność. W trzecim i czwartym, poziomy, przy których nastąpi włączenie (ON) i wyłączenie (Off) wentylatora. Wciśnięcie enkodera wywoła miganie trzeciego wiersza wyświetlacza umożliwiając ustawienie wartości wilgotności, przy której wentylator zostanie włączony. Kręcąc enkoderem można zwiększać i zmniejszać wartość parametru. Kolejne naciśnięcie enkodera pozwoli zmienić parametr określający poziom wilgotności, przy której nastąpi wyłączenie wentylatora. Program dba o to, aby pomiędzy poziomami włączenia i wyłączenia była różnica (histereza) co najmniej 10%. Trzecie naciśnięcie enkodera wyłącza ustawianie parametrów. Kręcąc enkoderem można wybrać inny ekran konfiguracyjny. Warto zaznaczyć, że pozostawienie aktywnego ekranu konfiguracyjnego lub zmiany parametru na dłużej niż minutę, spowoduje powrót do ekranu głównego.

Kolejnym ekranem jest ustawienie poziomu oświetlenia, który jest interpretowany

### Podstawowe parametry:

- pobór prądu ok 0,24 W (zasilanie 12 V 20 mA),
- max. obciążenie: 230 W (2 kW z radiatorem),
- parametry wyświetlane na wyświetlaczu oled,
- sterowanie przy pomocy enkodera przyciskiem,
- kryteria zał/wył: wilgotność, oświetlenie, manualnie,
- pasująca obudowa KM-50

### Projekty pokrewne na [www.media.avt.pl](http://www.media.avt.pl):

|          |  |
|----------|--|
| AVT-5698 | Sterownik wentylatorów 12 V dużej mocy (EP 8/2019)                   |
| AVT-5653 | Sterownik mikrowentylatora (EP 11/2018)                              |
| -        | Uniwersalny driver silnika małej mocy (EP 3/2018)                    |
| AVT-1981 | Sterownik wentylatora z płynną zmianą obrotów (EP 1/2018)            |
| AVT-5612 | Dwukierunkowy regulator obrotów silnika prądu stałego (EP 12/2017)   |
| AVT-5565 | Sterownik silnika do napędu (EP 10/2016)                             |
| AVT-478  | Regulator obrotów wentylatorów 12 V (EP 8/2016)                      |
| AVT-1855 | Sterownik wentylatora z czujnikiem wilgotności powietrza (EP 5/2015) |
| AVT-3082 | Zasilacz - sterownik miniwertarki DC (Edw 1/2014)                    |
| AVT-1724 | Uniwersalny sterownik silników DC (EP 2/2013)                        |
| AVT-1596 | Regulator obrotów wentylatora (EP 10/2010)                           |
| AVT-1519 | Sterownik silnika do modeli RC (EP 4/2009)                           |
| AVT-1469 | Generator PWM - regulator mocy silnika DC (EP 8/2008)                |
| AVT-2871 | Bi-motor driver (Edw 7/2008)   |
| AVT-1444 | Dwukierunkowy regulator obrotów silników prądu stałego (EP 12/2006)  |
| AVT-1564 | Sterownik wentylatora 12 V (EP 8/2001)                               |

### Wykaz elementów:

#### Rezystory SMD 1206:

R4: 0,33 Ω  
 R7: 0 Ω  
 R24, R28, R30, R31: 1,5 Ω  
 R27: 4,7 kΩ  
 R10\*: 10 kΩ  
 R9, R16, R20, R22, R29: 10 kΩ  
 R19, R23, R25, R26: 22 Ω  
 R12: 39 Ω  
 R8, R11, R13..R15, R17, R18, R32..R35: 360 Ω  
 R1, R3: 70 kΩ  
 R36\*, R37\*: 470 kΩ

#### Rezystory SMD 1812:

R2, R38: 22 Ω

#### Kondensatory SMD1206:

C12, C14: 5,6 pF  
 C7, C11: 15 nF  
 C13\*, C15\*: 22 pF  
 C8, C16, C17, C18, C19, C20: 100 nF  
 C4: 470 pF  
 C10, C21: 10 μF

#### Kondensatory THT:

C1: 1 μF/400 V  
 C9: 10 nF/400 V  
 C2: 470 nF/400 V  
 C5, C6: 470 μF/10 V elektrolityczny  
 C3: 1000 μF/16 V elektrolityczny  
 C2: 470 nF/400 V elektrolityczny  
 C5, C6: 470 μF/10 V elektrolityczny

#### Półprzewodniki:

U7: STM32F103C8T6  
 U3: TSOP48XX  
 U1: MC34063ACD SO-8  
 U2: MOC3020/3021 DIP6  
 T1: BT136 TO-220  
 U6\*: DS9503 SO-6  
 U8\*: FM24CL16B-GTR SO-8  
 D2, D8: SS14 DO21  
 M1: DF06S  
 R21: FOTOREZYSTOR 10 kΩ  
 D6, D11: Dioda LED żółta SMD1206  
 D7, D12: Dioda LED czerwona SMD1206  
 D3, D4, D9: Dioda LED niebieska SMD1206  
 D5, D10: Dioda LED Zielona SMD1206  
 D1: C12V 1W THT  
 J5: OLED 0,96" 128x64 niebieski SPI  
 J6\*: OLED RGB 0,95 96x64 KAMAMI  
 U4\*: DS18B20

#### Pozostałe:

F1: Oprawa ZH32 wraz z szybkim bezp. 2 A  
 F2: PCT 100 mA  
 Q2\*: Kwar 8 MHz  
 Q1\*: Kwar 32768 Hz  
 L1: Dławik DL22-100  
 J11: Czujnik DTH11  
 U5\*: ESP-01  
 IMPULSATOR: Enkoder impulsowy z przyciskiem  
 JP1: Goldpin 2x2  
 JP2: Goldpin 1x2  
 JP3, JP4: 821-1-06-S1 + T812-1-06 + taśma FLAT  
 J13, J14: SN25-W3K  
 J1, J7: ARKX  
 S1: Micoswitch 5x7  
 TR1\*: TZ1.9VA/9 V  
 J10\*: USB-B-BV

\* - nie montować (patrz tekst)



Fotografia 7. Ekran ustawiania poziomu oświetlenia



Fotografia 8. Ekran ustawień czasów



Fotografia 9. Ekran wyłączenia automatycznej akcji



Fotografia 10. Wygląd ekranu w stanie spoczynku

jako włączenie światła oraz poziomu traktowanego jako wyłączenie oświetlenia (fotografia 7). Drugi wiersz pokazuje wartość odczytaną (po uśrednieniu) z przetwornika ADC. Ustawianie poziomu załączenia i wyłączenia przebiega tak samo jak w przypadku wilgotności. W kolejnym ekranie (fotografia 8) można ustawić czas (On) po którym, od wykrycia włączenia oświetlenia, nastąpi załączenie wentylatora oraz jak długo po zgaśnięciu światła (Off) wentylator ma jeszcze pracować. Ustawienia te zapobiegają niepotrzebnemu włączaniu się wentylatora gdy wchodzimy na chwilę do łazienki oraz odprowadzić parę czy wywietrzyć pomieszczenie po jego dłuższym użytkowaniu. Kolejny ekran (fotografia 9) pozwala wyłączyć automatyczną reakcję na oświetlenie i/lub wilgotność. Wyłączenie automatyki spowoduje, że ekran przypisany do wybranej opcji będzie przyciemniony. Ostatni ekran wyświetla informacje diagnostyczne. Ich znaczenie można poznać analizując kod źródłowy programu.

Po skonfigurowaniu można przejść do użytkownika sterownika. Wygląd ekranu w stanie spoczynku przedstawia fotografia 10. Włączenie oświetlenia sygnalizuje

```
Listing 1. Podstawowe definicje
#define DEF_TIM_FAN_MAN_MAX (10*60*60*1000) // 10 godzin
#define DEF_TIM_FAN_ON_MAN (30 * 60 * 1000) // krok 30 minut
#define DEF_TIMEOUT_MENU (60*1000)
#define DEF_TIM_DEMO (5 * 1000)
#define DEF_EEPROM_TIMEOUT (10 * 1000) // 10 sekund
```

```
Listing 2. Fragment kodu pokazujący sposób użycia DMA do komunikacji z wyświetlaczem
HAL_GPIO_WritePin( LCD_C_D_GPIO_Port, LCD_C_D_Pin, GPIO_PIN_SET );
HAL_GPIO_WritePin( LCD_CS_GPIO_Port, LCD_CS_Pin, GPIO_PIN_RESET );
ssd1306_busy = 1;
HAL_SPI_Transmit_DMA( &hspl1, ssd1306buffer, SSD1306_LCDHEIGHT * SSD1306_LCDWIDTH / 8 );
```

```
Listing 3. Kod procedury uruchamianej przerwaniem z DMA
//Handler przerwania DMA:
void DMA1_Channel13_IRQHandler(void)
{
    /* USER CODE BEGIN DMA1_Channel13_IRQn 0 */
    DMA_HandleTypeDef *hdma = &hdma_spi1_tx;
    uint32_t flag_it = hdma->DmaBaseAddress->ISR;
    uint32_t source_it = hdma->Instance->CCR;

    /* Transfer Complete Interrupt management */
    if (((flag_it & (DMA_FLAG_TC1 << hdma->ChannelIndex)) != RESET)
        && ((source_it & DMA_IT_TC) != RESET)){
        extern void ssd1306_DMA_End();
        ssd1306_DMA_End();
    }

    /* USER CODE END DMA1_Channel13_IRQn 0 */
    HAL_DMA_IRQHandler(&hdma_spi1_tx);
    /* USER CODE BEGIN DMA1_Channel13_IRQn 1 */
    /* USER CODE END DMA1_Channel13_IRQn 1 */
}
```

```
Listing 4. Dodatkowa funkcja wywoływana przerwaniem z DMA
void ssd1306_DMA_End(){
    // Timer przy wartosci 1 wywola „ssd1306_CS_End()”
    extern uint8_t TimLcdCs;
    TimLcdCs = 1;
}
```

```
Listing 5. Dekrementowanie wirtualnego timera w przerwaniu systemowym
void Irqims() {
    //----- dezaktywacja, przez DMA, strobu LCD -----//
    extern void ssd1306_CS_End();
    if ( TimLcdCs == 1 ) ssd1306_CS_End();
    if ( TimLcdCs ) TimLcdCs--;
}
```

symbol żarówki w lewym górnym rogu wyświetlacza, ponadto rozpocznie się odliczanie czasu do włączenia wentylatora. Gdy wentylator zostanie załączony odliczany będzie maksymalny dopuszczalny czas pracy (10 godzin). Po zgaszeniu światła wentylator będzie nadal pracował przez czas ustawiony w menu „Set Time”. Na wyświetlaczu będzie widać upływ czasu do wyłączenia wentylatora. Jeśli przyczyną załączenia wentylatora będzie wzrost wilgotności na ekranie zobaczymy symbol „Rh”. W każdej chwili można wyłączyć wentylator naciskając przycisk enkodera. Możliwe jest też ręczne jego włączenie. Wentylator jest załączany na określony czas, który można regulować enkoderem w zakresie od 30 minut do 10 godzin, z krokiem 30 minut. Regulacja możliwa jest przez 60 sekund od włączenia wentylatora.

### Program sterujący

W artykule zostaną opisane tylko wybrane fragmenty programu, pełne kody źródłowe dostępne są w materiałach dodatkowych. Definicjami z listingu 1 można regulować maksymalny czas pracy wentylatora, krok czasu w trybie ręcznym, czas nieaktywności użytkownika, po którym program wraca z ekranu

ustawień do ekranu głównego, czas wyświetlania napisu powitalnego oraz czas po, którym, od zakończenia regulacji, nastawy zostaną zapamiętane w pamięci nieulotnej.

Funkcje obsługi czujnika DHT11 zostały przeniesione z Arduino. W przeciwieństwie do AVR w ARM można wykonać obsługę czujnika bez potrzeby wawieszania przerwań. Oczywiście lepiej byłoby do tego celu użyć UART (protokół DHT11 jest oparty o 1-Wire) ale w tym zastosowaniu nie wydawało się to celowe. Nie modyfikowałem też nadużywanych w Arduino typów FLOAT na INT ponieważ ARM jest na tyle szybki, że czasy wykonywania operacji na FLOAT nie mają większego znaczenia, tym bardziej, że wykonywane są stosunkowo rzadko.

Interesujące może być emulowanie pamięci EEPROM w FLASH. Zastosowano bibliotekę dostarczoną przez STMicroelectronics. Używając biblioteki trzeba pamiętać, aby w IDE zmniejszyć ilość dostępnej pamięci FLASH. Aby biblioteka działała, przed jej użyciem należy odblokować dostęp do zapisu pamięci FLASH, umieszczając poniższy kod:  
 HAL\_FLASH\_Unlock();  
 EE\_Init();

Bardziej eleganckim rozwiązaniem byłoby odblokowywanie pamięci przed procedurą zapisu i blokowanie po jej zapisaniu.

Procedury obsługi wyświetlacza także zostały przeniesione z Arduino, z tym, że transmisja używa mechanizmu DMA (**listing 2**). Pojawia się problem zdjęcia stanu linii CS. Co prawda DMA może generować przerwanie, gdy wyśle połowę i wszystkie dane (**listing 3**). Niestety nie można po zakończeniu transferu od razu zdjąć stanu CS, ponieważ SPI jeszcze transmituje dane. Takie natychmiastowe zdjęcie stanu powodowało obciążenie 4...6 bitów danych. Aby tego uniknąć, przerwanie DMA wywołuje funkcję z **listingu 4**. Funkcja ta ustawia wirtualny timer „TimLcdCs”, który jest dekrementowany w przerwaniu systemowym co 1 ms (**listing 5**). W przerwaniu, gdy TimLcdCs osiągnie wartość 1, wywołana zostanie funkcja dezaktywująca stan linii CS. W `ssd1306_DMA_End()` przypisanie

zmiennej „TimLcdCs” wartości 1 jest trochę niebezpieczne. Może się zdarzyć, że zaraz po przerwaniu DMA wykona się przerwanie 1ms. W konsekwencji stan linii CS zostanie zdjęty zbyt szybko. Może to powodować niekontrolowane zachowanie się ostatnich ośmiu pikseli na wyświetlaczu. Ze względu na to, że szansa na wystąpienie tego zjawiska jest bardzo mała oraz to, że wyświetlacz odświeżany jest 10 razy na sekundę nie modyfikowałem kodu. Rozwiązania są dwa, albo w przerwaniu `ssd1306_DMA_End()` do zmiennej „TimLcdCs” wpisywać liczbę 2 albo lepsze – uruchomić dodatkowy timer odliczający  $\mu$ s, który odczeka wymagany czas i zdejmie stan linii CS wyświetlacza. Należy przyznać, że obsługa wyświetlacza z interfejsem I<sup>2</sup>C jest dużo prostsza, uruchamia się DMA i to wszystko.

Program obsługuje „UsageFault\_Handler”, „BusFault\_Handler”,

„MemManage\_Handler”, „HardFault\_Handler” i inne błędy sygnalizując je diodą ERROR. W programie można wprowadzić szereg zmian. Można oprogramować moduł karty Wi-Fi umożliwiając współpracę z systemem inteligentnego budynku. Pisanie obsługi Wi-Fi ułatwić może uruchomienie interfejsu USB lub UART1 dostępnego na J9. Można też dodać obsługę 1-Wire a co za tym idzie, pomiar temperatury (termometr w DHT11 nie jest zbyt dokładny). W programie, od reakcji na bezwzględne wartości wilgotności, lepiej sprawdziłby się algorytm reagujący na wzrost/spadek wilgotności w zadanym czasie. Program steruje tylko diodami RUN i ERR (Error). Można dodać obsługę diody AL (Alert) oraz OK. Alert mógłby informować o awarii czujnika DHT11, OK o poprawnej pracy wszystkich czujników.

SaS

sas@elportal.pl

REKLAMA

## E-prenumerata to:

- najszybszy dostęp do nowego wydania magazynu
- wygodne archiwum na [www.avt.pl](http://www.avt.pl)
- hipertekstowy spis treści i wyszukiwarka
- wbudowane linki – klikasz i jesteś na odpowiedniej stronie WWW

Chcesz otrzymywać dodatek  
Niezbędnik Elektronika?  
Zamów prenumeratę  
drukowaną na  
[www.avt.pl/prenumerata](http://www.avt.pl/prenumerata)

Zamów e-prenumeratę (.pdf) na  
[www.avt.pl/prenumerata/elektroniczne](http://www.avt.pl/prenumerata/elektroniczne)

e-prenumerata roczna z rabatem  
**15% – 91,80 zł**

e-prenumerata dwuletnia z rabatem  
**30% – 151,20 zł**



Prenumeratory wersji drukowanej za równoległe e-wydania płać tylko 20% ceny: 21,60 zł/rok i 43,20 zł/2 lata