

# Moduły rozszerzeń dla Raspberry Pi (4)

**AVT  
5431**

Komputerek Raspberry Pi staje się coraz popularny, zakres jego aplikacji rośnie niemal w tempie, geometrycznym. Oprócz typowych funkcji multimedialnych coraz częściej budowane są różne aplikacje sterujące, między innymi do zastosowania w automatyce domowej.

Dla ułatwienia ich wykonywania opracowano szereg modułów rozszerzeń. Również my mamy propozycję dla użytkowników

Raspberry Pi.

**Rekomendacje:** płytki rozszerzają możliwości aplikacji Raspberry Pi, ułatwiają jego zastosowanie w układach sterujących. RaspPI\_LCD.

Płytkę interfejsu użytkownika



## RaspPI\_LCD. Płytkę interfejsu użytkownika

Przedstawiona płytkę umożliwia rozszerzenie funkcjonalności GPIO Raspberry Pi

### Wykaz elementów

#### Rezystory:

R1: 2,2 kΩ (SMD 0805)  
R2...R6: 330 Ω (SMD 0805)  
RV1: 10 kΩ

#### Kondensatory:

C1, C3: 100 nF (SMD 0805)  
C2: 10 μF (SMD 0805)

#### Półprzewodniki:

U2: DS1307 (SO8)

#### Inne:

BAT: bateria CR1220 z podstawką KEYS3000  
GPIO: złącze żeńskie IDC26  
I2C, S3V: złącze szpilkowe SIP4 2,54 mm  
LCD: wyświetlacz LCD 2×16 np. COG LCD-AC-C1602A-YGN NO/-E6 PBF  
LD: dioda LED 3 mm  
RES: złącze żeńskie SIP2  
SW1...SW5, RST: mikroprzycisk 6×3 mm  
XT: 32768 kHz (kwarce zegarkowy)

o możliwość pokazywania komunikatów na wyświetlaczu LCD zgodnym z HD44780, pięcioprzyciskową klawiaturę, zegar czasu rzeczywistego RTC oraz wyprowadzenie magistrali I<sup>2</sup>C i interfejsu szeregowego.

Schemat płytki rozszerzeń RaspPI\_LCD pokazano na **rysunku 1**. Jest ona kompatybilna z Raspberry Pi Rev2 (512 MB) o rozmieszczeniu wyprowadzeń GPIO pokazanym w **tabeli 1**.

Jako wyświetlacz zastosowano tani moduł alfanumerycznego wyświetlacza LCD o rozdzielczości 16×2 znaki wykonany w technologii COG, co pozwala zmniejszyć wymiary płytki. Wyświetlacz LCD jest zasilany z napięcia +5 V. Jest sterowany interfejsem 4-bitowym tylko w trybie zapisu. Co prawda wyświetlacz jest zasilany z 5 V, ale akceptuje sygnały sterujące o poziomie 3,3 V upraszczając interfejs. Potencjometr RV1 umożliwia ustawienie kontrastu. Płytkę wyposażono w 5-przyciskową klawiaturę (SW1...SW5). Rezystory ograniczają prąd w wypadku po-

### W ofercie AVT\*

AVT-5431 A

#### Podstawowe informacje:

- Napięcie zasilania 5 V DC.
- Wyświetlacz 2×16 znaków.
- Klawiatura z 5 przyciskami.
- Wbudowany zegar RTC.
- Kompatybilny z RaspberryPI ver. 2.

#### Dodatkowe materiały na CD lub FTP:

<ftp://ep.com.pl>, user: 08252, pass: 852rja63

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

#### Projekty pokrewne na CD/FTP:

(wymienione artykuły są w całości dostępne na CD)

AVT-5412,-13,-14	Moduły rozszerzeń dla Raspberry Pi (3) – RaspbPI_DIO16, RaspbPI_HUB, RaspbPI_DCM (EP 9/2013)
AVT-5402_2	Moduły rozszerzeń dla Raspberry Pi (2) – Płytkę do komunikacji szeregowej (EP 7/2013)
AVT-5402	Moduły rozszerzeń dla Raspberry Pi (1) – Płytkę stykową, moduł I/O, moduł wejść analogowych (EP 6/2013)

#### \* Uwaga:

Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytkę drukowaną PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx A+ płytkę drukowaną i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytkę drukowaną (lub płytki) oraz komplet elementów wymienionych w załączniku pdf  
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wmontowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można pobrać, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz (UK, A, A+, B lub C). <http://sklep.avt.pl>

myłkowegoysterowania GPIO w trybie wyjścia i zwarciu któregoś z przycisków. Na płytce zamontowano również zegar czasu rzeczywistego DS1307 lub DS1338-33 (U2). Układ współpracuje z podtrzymującą baterią

Tabela 1. Przyporządkowanie sygnałów GPIO1 – P1 (kolorem szarym zaznaczono wyprowadzenia używane przez RaspPI\_LCD)

Pin	Funkcja	Pin
P1-01	3.3V	P1-02
P1-03	GPIO2(SDA)	P1-04
P1-05	GPIO3(SCL)	P1-06
P1-07	GPIO4(RS)	P1-08
P1-09	GND	P1-10
P1-11	GPIO17(E)	P1-12
P1-13	GPIO27(D4)	P1-14
P1-15	GPIO22(D5)	P1-16
P1-17	3.3V	P1-18
P1-19	GPIO10(S1)	P1-20
P1-21	GPIO9(S2)	P1-22
P1-23	GPIO11(S3)	P1-24
P1-25	GND	P1-26

litowa typu CR1220 i typowym kwarem zegarkowym. Zwora RTC umożliwia wybór napięcia zasilania U2 pomiędzy 5 V dla DS1307 lub 3,3 V dla zgodnego DS1338-33.

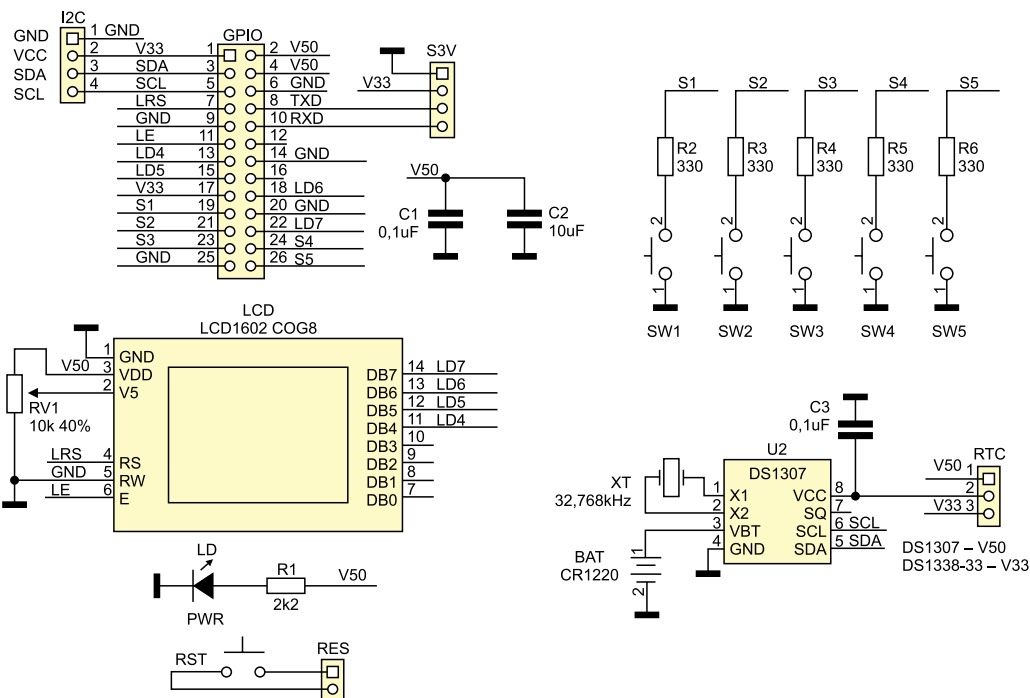
Dioda PW sygnalizuje załączenie zasilania, kondensatory C1...C3 filtrują zasilanie, przycisk RST umożliwia restart Rasperry. Dodatkowo, płytka jest wyposażona w złącza rozszerzeń I<sup>2</sup>C i S3V umożliwiające wyprowadzenie zgodne z Arduino Bricks magistrali I<sup>2</sup>C i interfejsu szeregowego Raspberry PI. **Uwaga!** Sygnały GPIO mają napięcie 3,3 V. Doprowadzenie wyższego napięcia może spowodować uszkodzenie GPIO.

Schemat montażowy RaspPI\_LCD pokazano na **rysunku 2**. Montaż jest typowy i nie wymaga opisu. Aby zapewnić stabilność mechaniczną i odciążyć delikatne wyprowadzenia LCD, przed przylutowaniem warto przykleić go do płytki kawałkiem samoprzylepnej, piankowej taśmy dwustronnej.

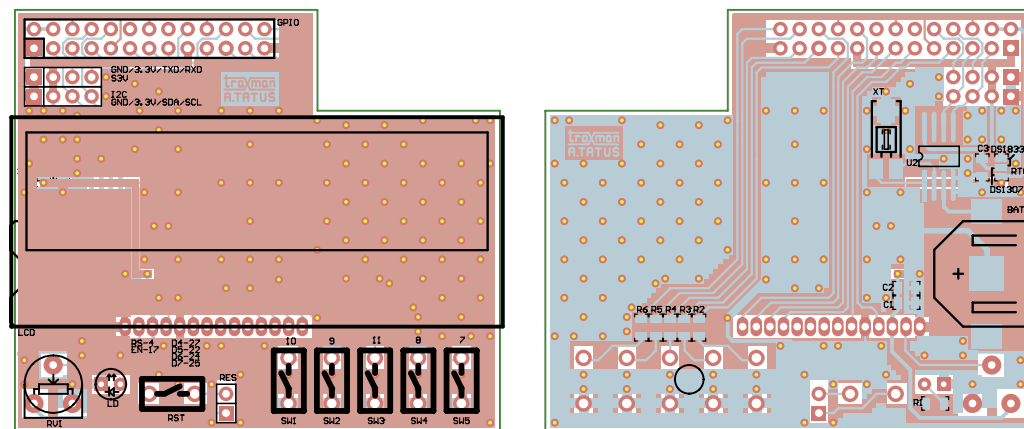
Uruchomienie układu wymaga skonfigurowania RasperryPI. Należy sprawdzić czy mamy aktualną wersję oprogramowania i wykonać ewentualną aktualizację do najnowszej wersji. W celu obsługi RTC jest konieczne dodanie obsługi interfejsu I<sup>2</sup>C. W tym celu sprawdzamy czy w pliku `sudo nano /etc/modules` znajduje się definicja `i2c-dev`. Jeżeli nie, to musimy ją dodać, zapisać zmiany i zrestartować PI. Po uruchomieniu należy pobrać narzędzia odpowiadające za obsługę I<sup>2</sup>C:

```
sudo apt-get install python-smbus
sudo apt-get install i2c-tools
Po zainstalowaniu, w pierwszej kolejności sprawdzamy w konsoli prawidłowe działanie I2C:
```

`sudo i2cdetect -y 1`  
Powinno pojawić się urządzenie pod adresem 0x68, jest to nasz RTC – DS1307. Następnie ładujemy moduł zegara:  
`sudo modprobe rtc-ds1307`  
`sudo bash echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device`  
Ustawienia czasu i daty systemowej dokonujemy poleceniem `sudo date`. Zapis czasu systemowego do RTC wykonujemy poleceniem `sudo hwclock -w`, a sprawdzenie poprawności zapisu `sudo hwclock -r`. Aby po uruchomieniu PI czas systemowy był automatycznie aktualizowany z RTC musimy dodać w pliku `sudo nano /etc/modules` linię



Rysunek 1. Schemat ideowy modułu RaspPI\_LCD



Rysunek 2. Schemat montażowy modułu RaspPI\_LCD

`rtc-ds1307` i w pliku `sudo nano /etc/rc.local` dodać linię:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
```

przed poleceniem `exit 0`. Przy kolejnym uruchomieniu PI, czas zostanie pobrany z RTC bez synchronizacji z zegarem sieciowym.

Obsługę klawiatury można sprawdzić tak jak w opisie wcześniejszych modułów z WebIOPi. Opis obsługi wyświetlacza zależy od preferowanego języka Python/C lub WiringPI. Dokładne opisy można znaleźć w sieci na Github oraz stronach poświęconych PI np. <http://goo.gl/PdPGIS> lub <http://goo.gl/ZLcRFS>.

### RaspbPI\_Relay. Płytką przekaźników

Płytką umożliwia rozszerzenie funkcjonalności GPIO Rasperry PI o dodatkowe dwa wyjścia przekaźnikowe i cztery wejścia cyfrowe z optoizolacją. Idealnie nadaje się do małego sterowania-monitorowania

**W ofercie AVT\***  
**AVT-5432 A**

**Podstawowe informacje:**

- Zasilanie 5 V DC.
- 4 przekaźniki, złącza śrubowe.
- Napięcie przełączane do 24 V DC.
- Kompatybilna z RasperryPI ver. 2.

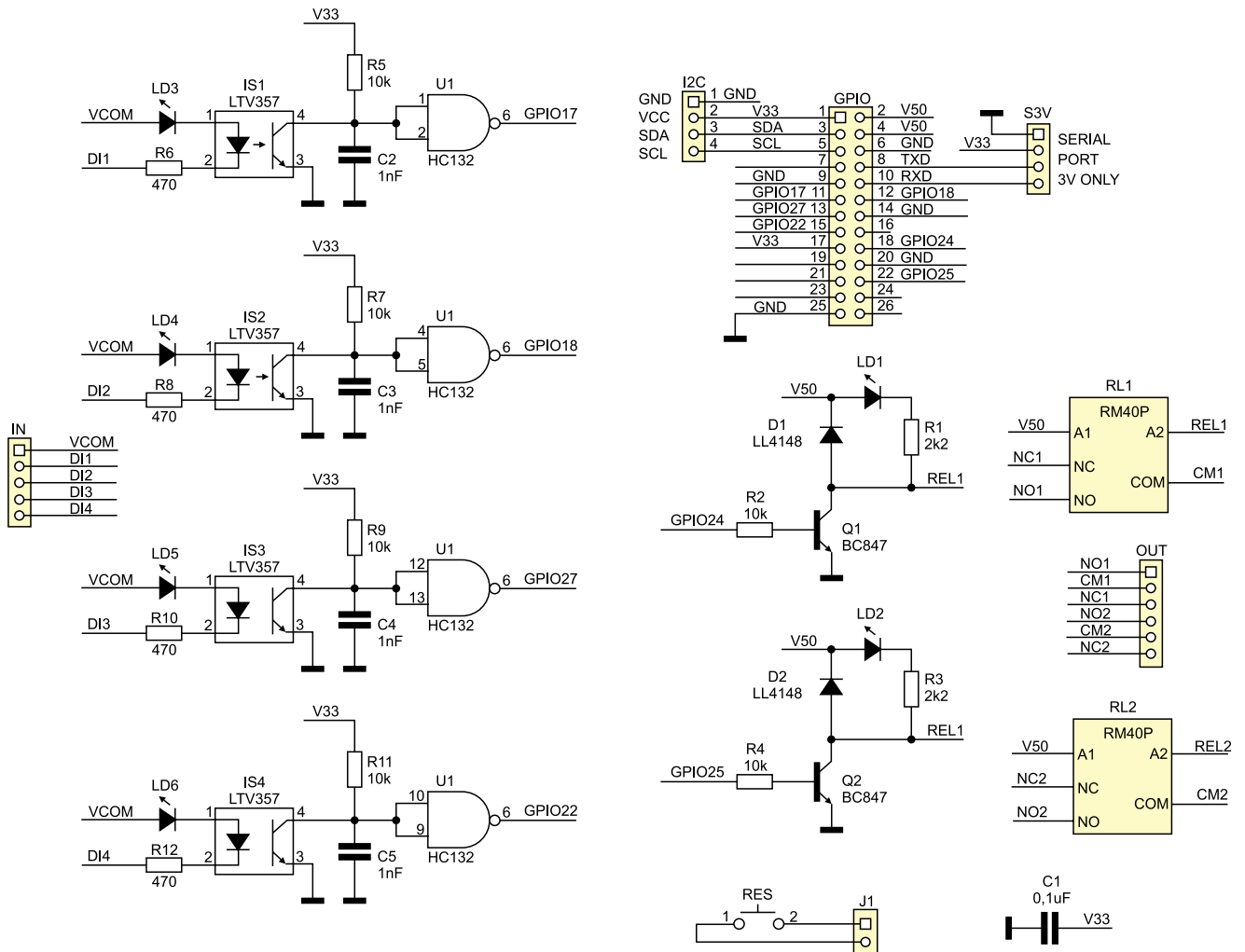
**Dodatkowe materiały na CD lub FTP:**  
<ftp://ep.com.pl>, user: 08252, pass: 852rja63

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

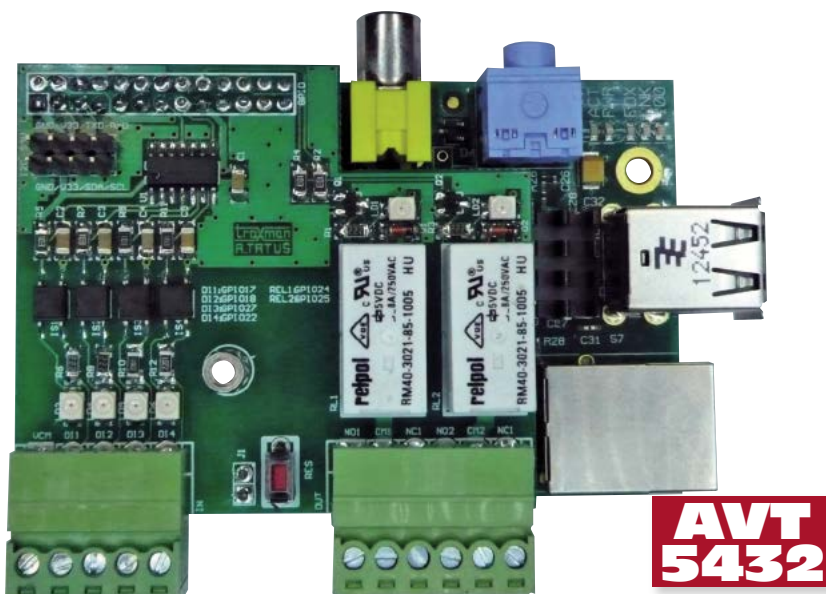
\* Uwaga:  
Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf  
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlotowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)  
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

poprzez WWW lub dla układów domowej automatyki.

Płytką jest kompatybilna z Rasperry Pi Rev2. o przyporządkowaniu sygnałów GPIO-P1 zamieszczonym w tabeli 1.



Rysunek 3. Schemat ideowy RaspPI\_Relay



#### Wykaz elementów

##### Rezystory:

R2, R4, R5, R7, R9, R11: 10 kΩ  
 R6, R8, R10, R12: 470 Ω  
 R1, R3: 2,2 kΩ

##### Kondensatory:

C1: 100 nF (SMD 1206)  
 C2...C5: 1 nF (SMD 1206)

##### Półprzewodniki:

D1, D2: LL4148 (diody uniwersalne SMD)  
 IS1...IS4: LTV357 (transpory SMD)  
 LD1...LD6: diody LED SMD, 2 mA  
 Q1, Q2: BC847 (SOT-23)  
 U1: HC132 (SO-14)

##### Inne:

GPIO: złącze IDC26, przelotowe  
 I2C, S3V: złącze szpilkowe, męskie  
 IN, OUT: złącze śrubowe  
 J1: złącze męskie SIP2  
 OUT: złącze śrubowe kompletne  
 RES: przycisk 6×3 mm  
 RL1, RL2: przekaźnik RM40P-5V

Schemat ideowy płytki z przekaźnikami pokazano na **rysunku 2**. Sygnały wejściowe są doprowadzone do złącza IN, wejścia współpracują z sygnałami czujników w konwencji NPN, które dla poprawnej pracy wymagają zasilania 18...32 V (VCOM). Jeżeli jest konieczne uzyskanie innego zakresu napięcia sterującego, wystarczy zmienić wartość rezystorów ograniczających prąd

dla zapewnienia ok. 10 mA dla diod LED transpotorów. Aktywne wejście sygnalizuje poziom świeceniem odpowiedniej diody LED.

Elementy RC na wejściu bramki NAND z wejściem Schmidta, odpowiadają za odfiltrowanie zakłóceń sygnałów doprowadzonych do GPIO. Przekaźniki sygnałów wyjściowych są sterowane są poprzez klucze

tranzystorowe Q1...Q2. Diody LD1...LD2 sygnalizują zasilanie cewki. Styki przekaźników wyprawodzone są na złącze OUT. **Uwaga!** Układ i współpracujące RaspberryPI nie są przystosowane do napięć sieci 230 V AC. Ze względu na bezpieczeństwo, do sterowania układów zewnętrznych zaleca się napięcie bezpieczne o wartości nieprzekraczającej 24 V AC/DC.

Listing 1. Skrypt relay.py do sterowania przekaźnikami

```

# Imports
import webiopi
# Enable debug output
webiopi.setDebug()
# Retrieve GPIO lib
GPIO = webiopi.GPIO
DI1 = 17
DI2 = 18
DI3 = 27
DI4 = 22
REL1 = 24
REL2 = 25
# Called by WebIOpi at script loading
def setup():
    # Setup GPIOs
    GPIO.setFunction(DI1, GPIO.IN)
    GPIO.setFunction(DI2, GPIO.IN)
    GPIO.setFunction(DI3, GPIO.IN)
    GPIO.setFunction(DI4, GPIO.IN)
    GPIO.setFunction(REL1, GPIO.OUT)
    GPIO.setFunction(REL2, GPIO.OUT)

while True:
    # Reading input
    DI1value = GPIO.digitalRead(DI1)
    DI2value = GPIO.digitalRead(DI2)
    DI3value = GPIO.digitalRead(DI3)
    DI4value = GPIO.digitalRead(DI4)
    # Show value
    print(„DI1: „ +str(DI1value)+” DI2: „ +str(DI2value)+” DI3: „
+str(DI3value)+” DI4: „ +str(DI4value))
    # D1/2, D3/4 set/reset RL1/2
    if (DI1value == True):
        GPIO.digitalWrite(REL1, GPIO.HIGH)
        print(„REL1: ON „)
    if (DI2value == True):
        GPIO.digitalWrite(REL1, GPIO.LOW)
        print(„REL1: OFF „)
    if (DI3value == True):
        GPIO.digitalWrite(REL2, GPIO.HIGH)
        print(„REL2: ON „)
    if (DI4value == True):
        GPIO.digitalWrite(REL2, GPIO.LOW)
        print(„REL2: OFF „)
    webiopi.sleep(2)
def destroy():
    webiopi.debug(„RELAY GPIO restore“)
    # Reset GPIO functions
    GPIO.setFunction(DI1, GPIO.IN)
    GPIO.setFunction(DI2, GPIO.IN)
    GPIO.setFunction(DI3, GPIO.IN)
    GPIO.setFunction(DI4, GPIO.IN)
    GPIO.setFunction(REL1, GPIO.IN)
    GPIO.setFunction(REL2, GPIO.IN)
    
```

Układ uzupełnia przycisk RES umożliwiający restart RaspberryPI. Dodatkowo, na złącza SIP wyprowadzone są sygnały magistrali I<sup>2</sup>C i interfejsu szeregowego dla zewnętrznych modułów rozszerzeń z wyprowadzeniami zgodnymi z Arduino oraz modułami I<sup>2</sup>C opisywanymi w Elektronice Praktycznej. Sygnały są zgodne z logiką napięciową 3,3 V.

Tabela 1. Przyporządkowanie sygnałów GPIO1 – P1 (kolorem szarym zaznaczono wyprowadzenia używane przez RaspbPI\_Relay)

Pin	Funkcja	Pin
P1-01	3.3V	P1-02
P1-03	GPIO2(SDA)	P1-04
P1-05	GPIO3(SCL)	P1-06
P1-07	GPIO4	P1-08
P1-09	GND	P1-10
P1-11	GPIO17(DI1)	P1-12
P1-13	GPIO27(DI3)	P1-14
P1-15	GPIO22(DI4)	P1-16
P1-17	3.3V	P1-18
P1-19	GPIO10(MOSI)	P1-20
P1-21	GPIO9(MISO)	P1-22
P1-23	GPIO11(SCLK)	P1-24
P1-25	GND	P1-26

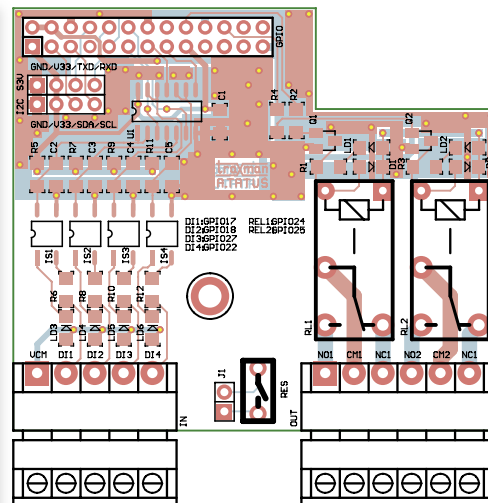
Schemat montażowy modułu RaspPI\_Relay pokazano na **rysunku 4**. Montaż jest typowy i nie wymaga opisu. Dla zapewnienia stabilności mechanicznej zaleca się zamocowanie płytki z przekaźnikami do RaspberryPI za pomocą tulejki dystansowej M3×10(12), zapobiega to wyłamywaniu złącza GPIO podczas manipulacji przy złączkach IN/OUT.

Dla szybkiego sprawdzenia modułu polecam użycie WebIOpi (<http://goo.gl/y2Wty5>). Oprogramowanie pobiera i instaluje się za pomocą poleceń:

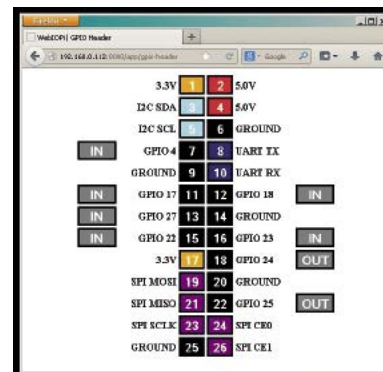
```

wget http://webiopi.googlecode.com/files/WebIOpi-0.6.0.tar.gz
tar xvfz WebIOpi-0.6.0.tar.gz
cd WebIOpi-0.6.0
sudo ./setup.sh
    
```

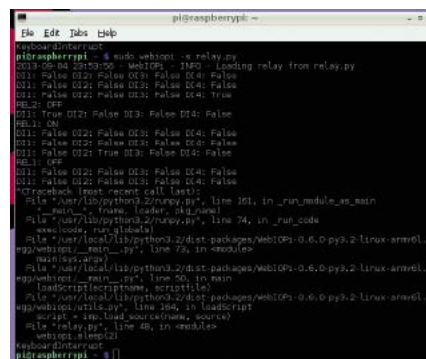
**Uruchomienie serwera** `sudo /etc/init.d/webiopi start`. Po przejściu do przeglądarki internetowej pod adres `http://localhost:8000/` (użytkownik: `webiopi`, hasło: `raspberry`) lub z innego komputera `http://<adres IP>:8000/`, wybraniu linku GPIO Header, ustawieniu trybu pracy pinów GPIO 17, 18, 22, 27 jako wejścia, a GPIO 24, 25 jako wyjścia, powinny być widoczne odczyty z wejść (**rysunek 5**) oraz powinna być możliwa zmiana stanu wyjść.



Rysunek 4. Schemat montażowy RaspPI\_Relay



Rysunek 5. Obsługa modułu przekaźników poprzez WebIOpi



Rysunek 6. Efekt działania skryptu z listingu 1

Moduł można wykorzystać znacznie ciekawiej korzystając z możliwości uruchamiania skryptów języka Python przez `webiopi`. W tym celu jest konieczne przygotowanie skryptu `relay.py` (**listing 1**). Realizuje on typowe w automatyce sterowanie stabilne SR cewkami przekaźników za pomocą dwóch chwilowych przycisków START i STOP. Wejścia DI1 i DI3 załączają przekaźniki REL1, REL2, wejścia DI2 i DI4 wyłączają. Stan wejść i przekaźników jest prezentowany za pomocą konsoli. W wypadku uruchamiania skryptu nie jest wymagane wcześniejsze uruchomienie serwera `webiopi`, zostanie on każdorazowo uruchomiony automatycznie.

W celu uruchomienia skryptu `relay.py` w terminalu wpisujemy komendę `$ sudo webiopi -s relay.py`. Uruchomiony skrypt można przerwać klawiszami CTRL+C. Efekt działania skryptu pokazano na **rysunku 6**. Sprawdzony, uruchomiony i oprogramowany moduł gotowy jest do przejęcia kontroli nad światem lub przynajmniej internetowym ekspresem do kawy...

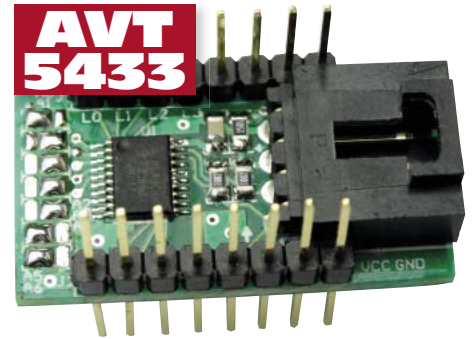
### LED8\_PWM\_Expander

*Przedstawiony moduł umożliwia rozszerzenie możliwości płytek uruchomionych, szczególnie Raspberry PI wyposażonego tylko w jeden kanał PWM. Dodatkowe 8 kanałów PWM o rozdzielczości 8 bitów jest przeznaczonych głównie do sterowania diodami LED RGB(Y), ale nadają się także do innych aplikacji np. sterowania prędkością obrotową silników DC.*

Schemat ideowy ekspandera zaprezentowano na **rysunku 7**. Sercem modułu jest układ PCA9634, jego schemat blokowy przedstawia **rysunek 8**. PCA9634 wchodzi w skład szerokiej rodziny kontrolerów LED z interfejsem I<sup>2</sup>C firmy NXP. Układ ma możliwość ustalania adresu za pomocą wyprowadzeń A0...A6. Dostępne jest 126 adresów I<sup>2</sup>C, kilka adresów jest zarezerwowanych dla sterowania globalnego.

Adres 0x03 jest zarezerwowany dla instrukcji zerowania, natomiast adres 0x70 dla jednoczesnego sterowania wszystkimi kanałami PWM wszystkich układów PCA9634 dołączonych do szyn I<sup>2</sup>C. Ułatwia to oprogramowanie układów wyświetlaczy składających się z kaskady PCA9634, gdyż wszystkie układy – niezależnie od adresu fizycznego – reagują na wspólną komendę, co znacząco odciąża interfejs I<sup>2</sup>C i ułatwia sterowanie. Możliwe jest także definiowanie adresów podrzędnych, grupujących sterowane kanały PWM, ułatwiających np. tworzenie ekranów z diod LED RGB.

Oprócz indywidualnego sterowania przebiegiem PWM na każdym z wyjść jest możliwe sterowanie grupowe (np. w celu regulowania jasności świecenia wszystkich diod LED) oraz praca przerywana z regulowaną częstotliwością i wypełnieniem. Każde z wyjść może być ustawione lub wyzerowane, jest możliwa sprzętowa negacja wyjścia dla pracy z zewnętrznym buforem oraz konfiguracja wyjścia jako otwarty dren lub totem-pole. Wewnętrzny oscylator i dzielniki programowane upraszczają aplikację PCA9634. Układ ma wejście sprzętowe OE umożliwiające sterowanie buforem wyjściowym, które w modelu nie jest używane. Układ funk-



#### W ofercie AVT\* AVT-5433 A

##### Podstawowe informacje:

- Kompatybilny z Arduino, RaspberryPI i innymi.
- Układ scalony ekspandera PCA9634.
- Zasilanie 5 V DC.
- 8 wyjść PWM np. do sterowania diodami LED.
- Funkcjonalny interfejs I<sup>2</sup>C.

##### Dodatkowe materiały na CD lub FTP:

<ftp://ep.com.pl>, user: 08252, pass: 852rja63

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

\* Uwaga:  
Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf to nic innego jak zmontowany zestaw B, czyli elementy wstawiane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można pobrać, klikając w link umieszczony w opisie kitu).  
AVT xxxx CD Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

REKLAMA

Tabela 1. Rejestry konfiguracyjne PCA9634		
Adres	Nazwa	Funkcja
0	MODE1	Rejestr konfiguracji układu
1	MODE2	Rejestr konfiguracji buforów
2	PWM0	Rejestr PWM 0
3	PWM1	Rejestr PWM 1
4	PWM2	Rejestr PWM 2
5	PWM3	Rejestr PWM 3
6	PWM4	Rejestr PWM 4
7	PWM5	Rejestr PWM 5
8	PWM6	Rejestr PWM 6
9	PWM7	Rejestr PWM 7
0A	GRPPWM	Rejestr grupowego PWM
0B	GRPFREQ	Rejestr grupowego „migania”
0C	LEDOUT0	Konfiguracja trybu grupy LED0 (LD0..3)
0D	LEDOUT1	Konfiguracja trybu grupy LED0 (LD4..7)
0E	SUBADR1	Konfiguracja podadresu globalnego grupa 1
0F	SUBADR2	Konfiguracja podadresu globalnego grupa 2
10	SUBADR3	Konfiguracja podadresu globalnego grupa 3
11	ALLCALLADR	Konfiguracja adresu globalnego ALL CALL LED

Tabela 2. Najważniejsze bity konfiguracyjne PCA9634		
Adres	Nazwa rejestru	Funkcja Bitu
0	MODE1	B7, 1/0= autoinkrementacja rejestrów, adresowanie bezpośrednie B4, 0/1= praca/uśpienie Pozostałe domyślnie = 0
1	MODE2	B5 0= grupowe PWM. 1= grupowe „miganie” B4 1= negacja wyjścia B3 0/1= zmiana stanu wyjść I2C po STOP/ACK B2 0/1= wyjścia open-drain/totem-pole Pozostałe domyślnie = 0
2:9	PWM0:9	Indywidualne PWM 0:7 0x00..0xFF
0A	GRPPWM	Grupowe PWM 0x00..0xFF W trybie migania określa wypełnienie % GRPPWM/256
0B	GRPFREQ	Grupowe „miganie” 0x00..0xFF czas w sekundach (GRPFREQ+1)/24 [s]
0C/0D	LEDOUT0/1	Konfiguracja LED3..0/LED7..4 B7/6,B5/4,B3/2,B1/0: 00= LEDx OFF 01= LEDx ON 10= LEDx indywidualny PWMx 11= LEDx indywidualny PWMx z grupowym PWM/ „miganiem” w zależności od MODE2:B5

**Wykaz elementów**  
**Rezystory:** (SMD 0805)  
 R1: 2,2 kΩ  
 R2, R3: 10 kΩ  
**Kondensatory:**  
 C1: 100 nF (SMD 0805)  
**Półprzewodniki:**  
 U1: PCA9634PW (SSOP20)  
 LD: dioda LED SMD  
**Inne:**  
 A0...A6: 0 Ω (SMD 0805, zwora)  
 I2C: złącze EH4, kątowe  
 J1, J2: złącze szpilkowe, męskie, 8-pin

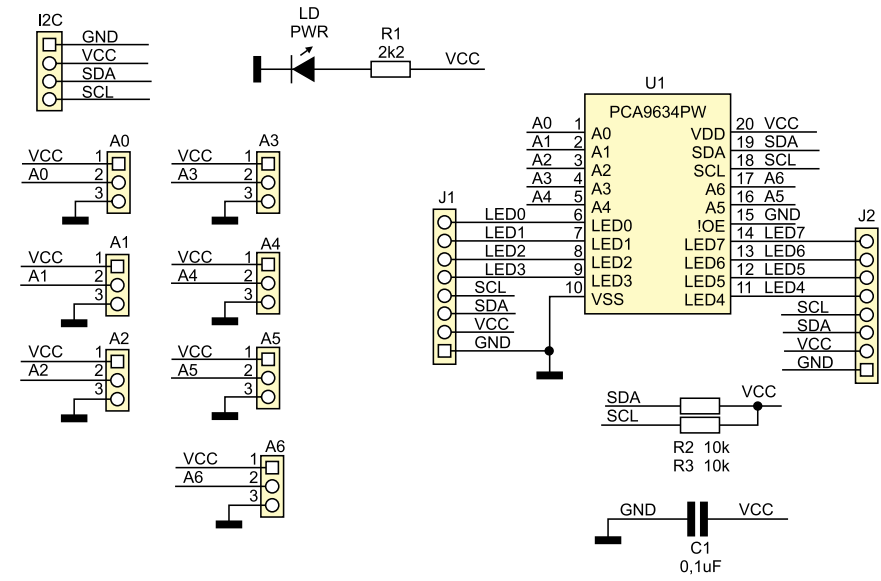
cjonuje poprawnie w zakresie napięcia zasilania 2,5...5,5 V.

Rejestry konfiguracyjne umieszczone w tabeli 1. W zależności od potrzeb jest możliwa komunikacja z poszczególnymi rejestrami poprzez adresowanie poszczególnych rejestrów (ze wskazaniem konkretnego rejestru, zgodnie z rysunkiem 9). Drugim sposobem jest adresowanie grupowe wszystkich rejestrów, jak pokazano na rysunku 10. Ostatnim sposobem jest adresowanie grupowe rejestrów PWM po wcześniejszej konfiguracji układu przedstawione na rysunku 11.

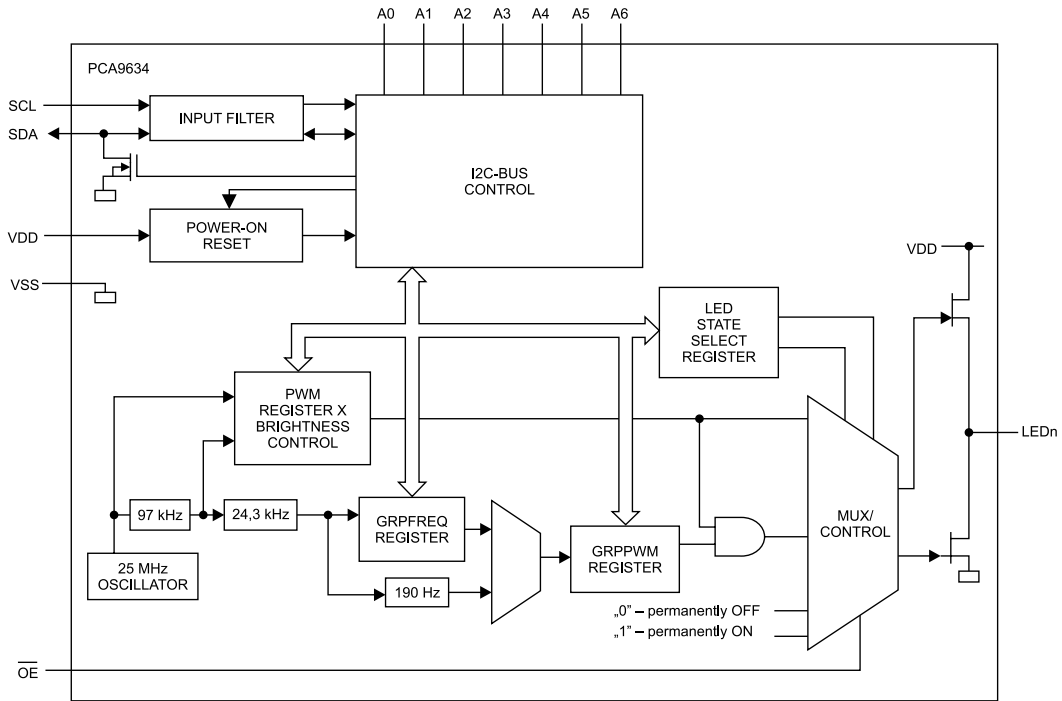
Odczyt odbywa się podobnie. Dokładniejsze informacje oczywiście zamieszczone są w nocie katalogowej. Najważniejsze bity konfiguracyjne dla podstawowego trybu pracy przedstawia tabela rys. 7.

Montaż układu jest typowy i nie wymaga opisu. Zastosowano płytkę dwustronną, rozmieszczenie elementów pokazano na rysunku 12. Wyprowadzenie I<sup>2</sup>C jest zgodne ze standardem Arduino. Pozostałe sygnały wyprowadzono na złącza szpilkowe o rastrze zgodnym z płytkami prototypowymi i stykowymi. Poprawnie zmontowany układ nie wymaga uruchamiania. Dioda PWR sygnalizuje załączenie zasilania. Przed uruchomieniem jest konieczne ustawienie adresu poprzez zlutowanie odpowiednich punktów A0...A6 ze zwróceniem uwagi na adresy zarezerwowane i ewentualne kolizje z innymi układami I<sup>2</sup>C. Prototyp miał ustawiony adres 0x61 (A0, A5, A6=VCC, reszta GND) i taki obsługuje przykładowy szkic.

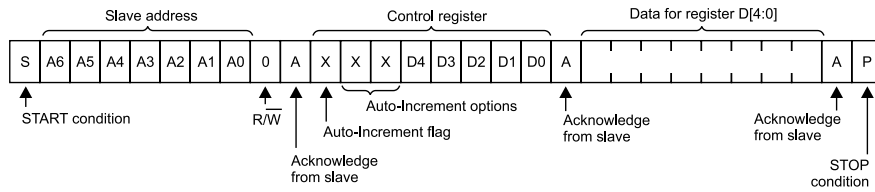
Dla szybkiego sprawdzenia podstawowych możliwości PCA9634 przygotowałem szkic dla Arduino/Energii (listing 2). Program konfiguruje rejestry PCA w trybie zapisu grupowego z automatyczną inkrementacją adresu. Następnie są konfigurowane kanały PWM. Pomiędzy wyjścia LEDx a masę układu są włączone diody świecące z rezystorami ograniczającymi prąd. Pętla kolejno zwiększa wypełnienie kanałów LED0...LED7. Następnie PCA jest konfigurowany do pracy z globalnym PWM jednocześnie ściemniając wszystkie kanały PWMx. Ostatnim krokiem jest konfiguracja dla trybu „migania” global-



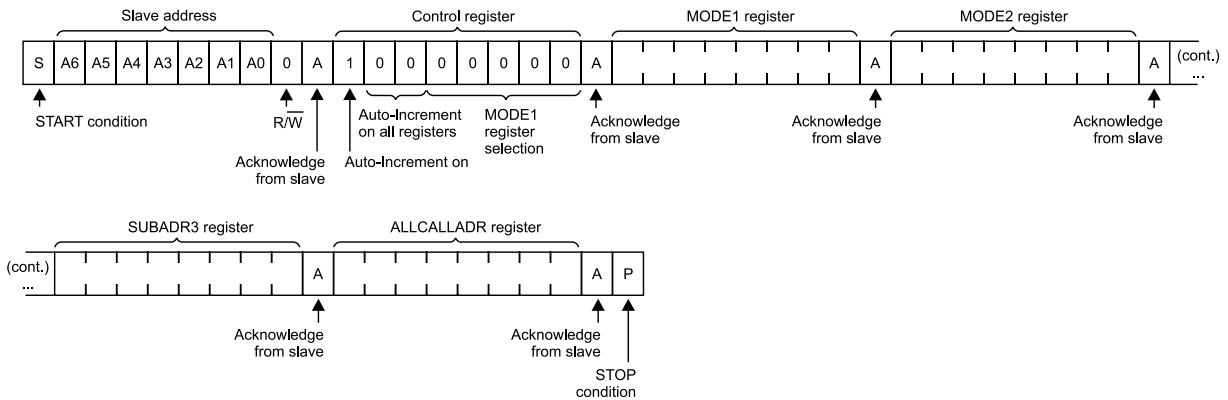
Rysunek 7. Schemat ideowy ekspandera PWM



Rysunek 8. Struktura wewnętrzna PCA9634 (za notą NXP)

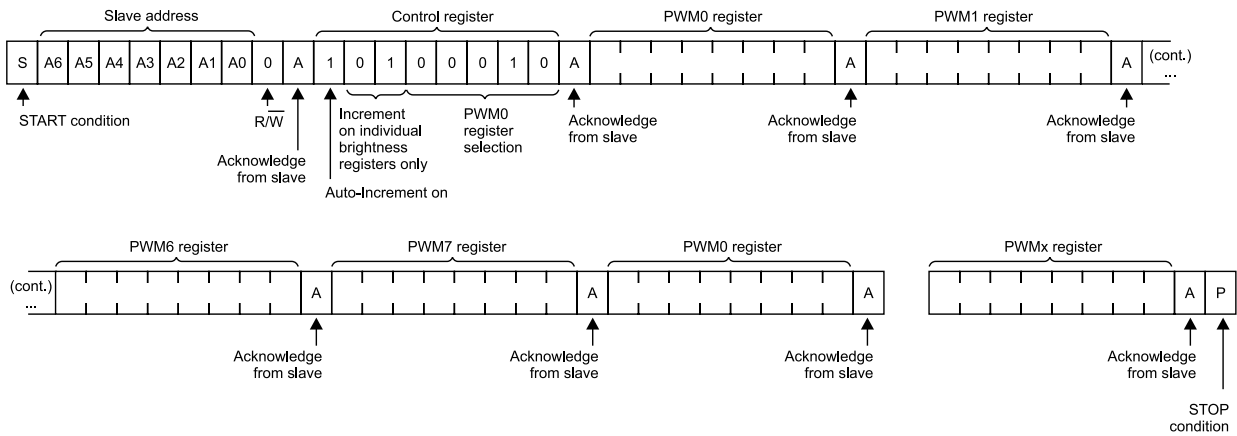


Rysunek 9. Adresowanie indywidualne rejestrów (za notą NXP)

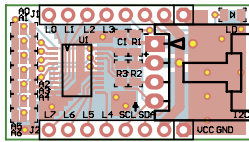


Rysunek 10. Adresowanie grupowe rejestrów (za notą NXP)

REKLAMA



Rysunek 11. Adresowanie grupowe rejestrów PWM (za notą NXP)



Rysunek 12. Schemat montażowy ekspandera PWM

nego, z ustawionym wypełnieniem 50% (GRPPWM) i czasem 1 s (GRPFREQ).

W przypadku współpracy z Raspberry PI najłatwiej sprawdzić poprawność pracy modułu korzystając z bibliotek i2c i konsoli. Oczywiście, nic nie stoi na przeszkodzie w napisaniu skryptu w Pythonie lub programu w C, ale te kilka komend dla szybkiego sprawdzenia można wydać z linii poleceń. W celu użycia PCA9634 jest konieczne dodanie obsługi magistrali I<sup>2</sup>C (jeżeli nie zrobiono tego wcześniej). W tym celu sprawdzamy czy w pliku *sudo nano /etc/modules* znajduje się definicja *i2c-dev*. Jeżeli nie, to musimy ją dodać, zapisać zmiany i zrestartować Raspberry PI. Po uruchomieniu należy pobrać narzędzia odpowiadające za obsługę I<sup>2</sup>C:

```
sudo apt-get install python-serial
sudo apt-get install i2c-tools
```

Po zainstalowaniu i restarcie, w pierwszej kolejności sprawdzamy w konsoli prawidłowe działanie I<sup>2</sup>C *sudo i2cdetect -y 1*. Powinno pojawić się urządzenie pod adresem 0x61 (lub innym ustawionym zworkami A6...0) – jest to nasz PCA9634. Jeżeli zdefiniowane zostały adresy globalne i została uaktywniona funkcja ich wykorzystania, powinny być także widoczne. Korzystając z polecenia *sudo i2cset -y 1 Adres\_I2C, Adres\_Rejestru, Dana\_Do\_Zapisu* możemy kolejno skonfigurować rejestry PCA np. w celu sprawdzenia działania układu. Po pomyślnych testach, moduł można zastosować we własnej aplikacji.

Adam Tatuś, EP

```
Listing 2. Program testowy dla PCA9634, dla Arduino/Energia
/*****
 * TEST PCA9634 LED8 PWM Expander ADR=61
 *****/

#include <Wire.h>
#define adr 0x61 //A6,5,0=VCC, reszta GND

void setup()
{
  byte data[19]={ // konfiguracja PCA, zapis grupowy rejestrów
    0x80, // Autoinkrementacja adresu
    0x80,0x14, // MODE1/2 bez subadresow, grupowe sciemnianie, opendrain,
    inversja
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, //PWMx = off
    0x80,0x40, //PWM 50%, BLINK 1.8s
    0xAA,0xAA, //LEDOUTx indywidualny PWMx
    0xE2,0xE4,0xE8,0xE0 }; // domyślne podadresy globalne

  Wire.begin(); // inicjacja I2C
  delay(1);
  Wire.beginTransmission(adr);
  Wire.write(data, 19); // konfigurowanie PCA
  Wire.endTransmission();
}

void loop()
{
  Wire.beginTransmission(adr);
  Wire.write(0x00);
  Wire.write(0b00000000); // MODE1
  Wire.endTransmission();
  Wire.beginTransmission(adr);
  Wire.write(0x01);
  Wire.write(0b00011100); // MODE2, dim
  Wire.endTransmission();
  //PWM indywidualny LEDx=PWMx
  for (int y=2; y<10;y++) //PWM0..7 min->max
  {
    for (int x=0; x<0xFF; x++)
    {
      Wire.beginTransmission(adr);
      Wire.write(y);
      Wire.write(x);
      Wire.endTransmission();
      delay(5);
    }
  }
  // PWM globalny GRPPWM max..min
  Wire.beginTransmission(adr);
  Wire.write(0x01);
  Wire.write(0b00011100); // MODE2
  Wire.endTransmission();
  Wire.beginTransmission(adr);
  Wire.write(0x0C); // LEDOUT0, global dim/pwm
  Wire.endTransmission();
  Wire.beginTransmission(adr);
  Wire.write(0x0D); // LEDOUT1, global dim/pwm
  Wire.endTransmission();
  for (int x=0xFF; x>0; x--) //GRPPWM min->max
  {
    Wire.beginTransmission(adr);
    Wire.write(0x0A);
    Wire.write(x); //GRPWM
    Wire.endTransmission();
    delay(20);
  }
  // Globalne miganie 1s/50% wypełnienia
  Wire.beginTransmission(adr);
  Wire.write(0x01);
  Wire.write(0b00011100); // MODE2, blink
  Wire.endTransmission();
  Wire.beginTransmission(adr);
  Wire.write(0x0A);
  Wire.write(0x80); //GRPWM 50% wypełnienia BLINK
  Wire.endTransmission();
  Wire.beginTransmission(adr);
  Wire.write(0x0B);
  Wire.write(0b00100000); // GRPFREQ = 1s
  Wire.endTransmission();
  while(1);
}
```