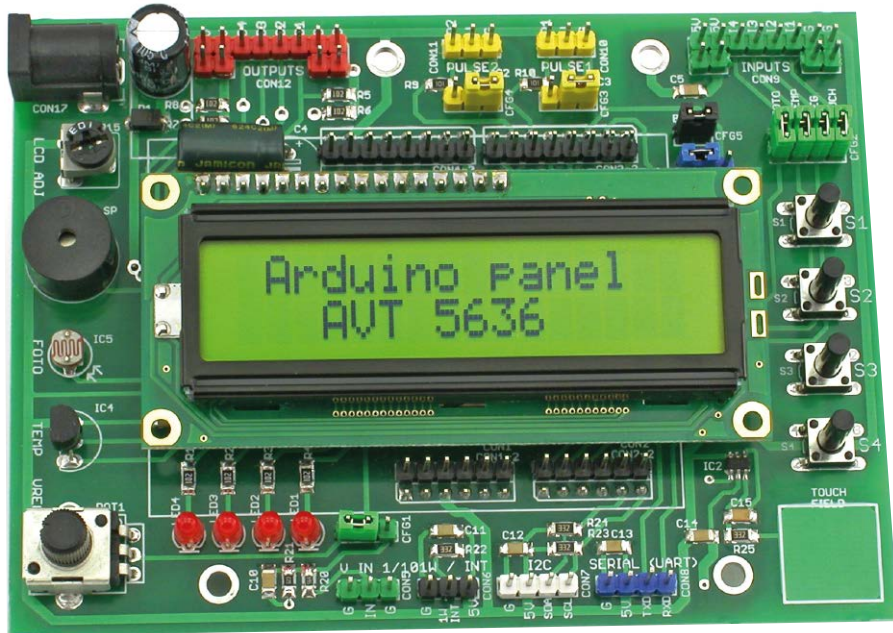


Płytki edukacyjna dla Arduino

Po zbudowaniu kilku urządzeń, których „silnikiem” jest płytki Arduino zauważamy, że zwykle wymagają one stałego zestawu komponentów dodatkowych, takich jak: wyświetlacz, przyciski czy diody LED. Każdorazowe uzbrajanie płytki w te same komponenty to strata czasu i okazja do popełnienia błędu. Prezentowana płytki znacznie usprawni pracę a dodatkowo stanowi doskonałą bazę do nauki i ćwiczeń z Arduino.

Rekomendacje: płytki przyda się do wykonywania prototypów urządzeń oraz osobom chcącym nauczyć się programowania mikrokontrolerów.



W trakcie pracy ze środowiskiem Arduino bardzo pomocny będzie schemat funkcjonalny z **rysunku 1**. Pokazano na nim uproszczony schemat połączeń pomiędzy płytką Arduino a komponentami na płytce edukacyjnej. W **tabelach 1 i 2** umieszczono opis wszystkich złączy konfiguracyjnych płytki.

Budowa i montaż

Dokładny schemat elektryczny pokazano na **rysunku 2**. Podzielono go na wiele małych bloków funkcyjnych, ułożonych niemal identycznie jak odpowiadające im elementy na płytce drukowanej. Schemat jest rozbudowany, ale nie jest skomplikowany i nie wymaga szczegółowego komentarza.

Schemat płytki drukowanej oraz rozmieszczenie elementów pokazano na **rysunku 3**. Montaż należy wykonać według ogólnych zasad. Złącza CON1...CON4 umieszczone są na płytce podwójnie - jedno należy zamontować standardowo a te opisane **ARD** należy zamontować od spodu płytki tak, żeby pasowały do złączy płytki Arduino. Warto zastosować szpilki goldpin dłuższe od klasycznych, dzięki temu nie będzie niebezpieczeństwa, że elementy płytki Arduino zetkną się z punktami lutowniczymi na płytce edukacyjnej.

Mikrokontroler IC1 sterujący pracą wyświetlacza najlepiej zaprogramować przed zamontowaniem na płytce (mikrokontroler z zestawu AVT5636 będzie już

zaprogramowany). Ostatecznie mikrokontroler można zaprogramować podłączając się do niego programatorem poprzez złącze wyświetlacza – są tam wyprowadzone sygnały MISO, MOSI i SCK, sygnał RST połączony jest z sygnałem RESET płytki Arduino, na złączu CON1.

Jedyną czynnością uruchomieniową, po zmontowaniu płytki, jest ustawienie kontrastu wyświetlacza. Do tego celu służy miniaturowy potencjometr oznaczony **LCD ADJ**. Najpierw należy zasilic płytkę edukacyjną dołączając napięcie do szpilek 5V i G dowolnego złącza lub po prostu dołączając i zasilając płytę Arduino. Sterownik wyświetlacza lcd, na płytce edukacyjnej, jeśli nie otrzyma żadnego polecenia w ciągu ok 2 sekund od włączenia zasilania to automatycznie wyświetli przykładowy powitalny komunikat. Wtedy należy regulować potencjometrem aż do uzyskania dobrze widocznej treści.

Zamiast Hello World

Zamiast programu testującego, wyświetlacza klasyczny komunikat, do płytki edukacyjnej została przygotowana dedykowana biblioteka dla środowiska programistycznego Arduino. Dzięki temu szybko i bez analizowania budowy płytki można wykorzystać jej możliwości. Wykaz wszystkich metod wraz z krótką charakterystyką zawarty jest w **tabeli 3**.

Wykaz elementów:

Rezystory: (SMD 1206)

R1...R8, R11...R14: 1 kΩ
R9, R10, R17: 100 Ω
R15: 10 kΩ (pot. miniaturowy)
R16: 47 Ω
R18, R19, R22...R26: 3,3 kΩ
R20: 62 kΩ
R21: 560 kΩ
POT1: potencjometr 10 kΩ typu RV091

Kondensatory:

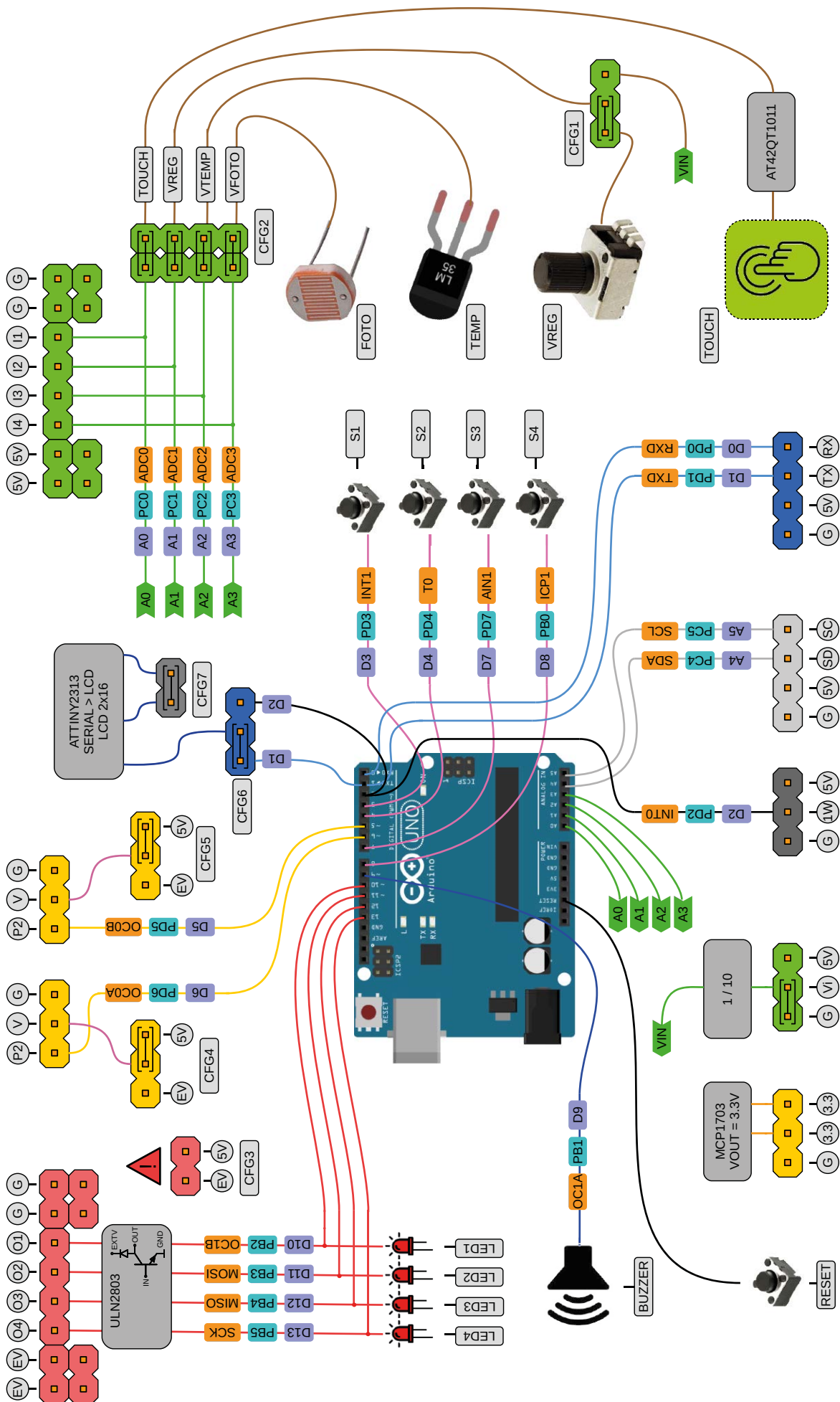
C1: 470 μF/25 V
C4: 1000 μF/6,3 V
C2, C3, C5, C6, C9...C14, C16...C18: 100 nF
C7, C8: 18 pF
C15: 10 nF

Półprzewodniki:

D1: 1N4007
D2: dioda Zenera 5,6 V
T1: BC847 (SOT23)
LED1...LED4: LED f 3 mm
IC1: ATtiny2313 (zaprogramowany)
IC2: AT42QT1011 (SO8)
IC3: ULN2803 (SO16)
IC4: LM35
IC5: fotorezystor
IC6: MCP1703-3,3 SOT223

Inne:

X1: rezonator 11,059200 MHz
SP: buzzer z generatorem
S1...S4, SW1: mikroprzycisk
LCD: wyświetlacz LCD2x16
CFG1...CFG7: goldpin + jumper
CON1...CON4: goldpin 6+6+8+8
CON1...CON4 ARD: goldpin wysoki 6+6+8+8
CON5...CON16: goldpin różne kolory
CON17: gniazdo DC2,1/5,5



Rysunek 1. Schemat funkcjonalny płytki edukacyjnej dla Arduino

Aby dołączyć bibliotekę do programu należy najpierw umieścić odpowiedni plik w katalogu, gdzie środowisko Arduino wyszukuje dodatkowych bibliotek. Jest to lokalizacja „c:\... \Documents\Arduino\libraries” i należy tam umieścić cały katalog biblioteki AVT5363lib. Po uruchomieniu środowiska

DODATKOWE MATERIAŁY DO POBRANIA ZE STRONY:

www.media.avt.pl

W ofercie AVT* AVT-5636

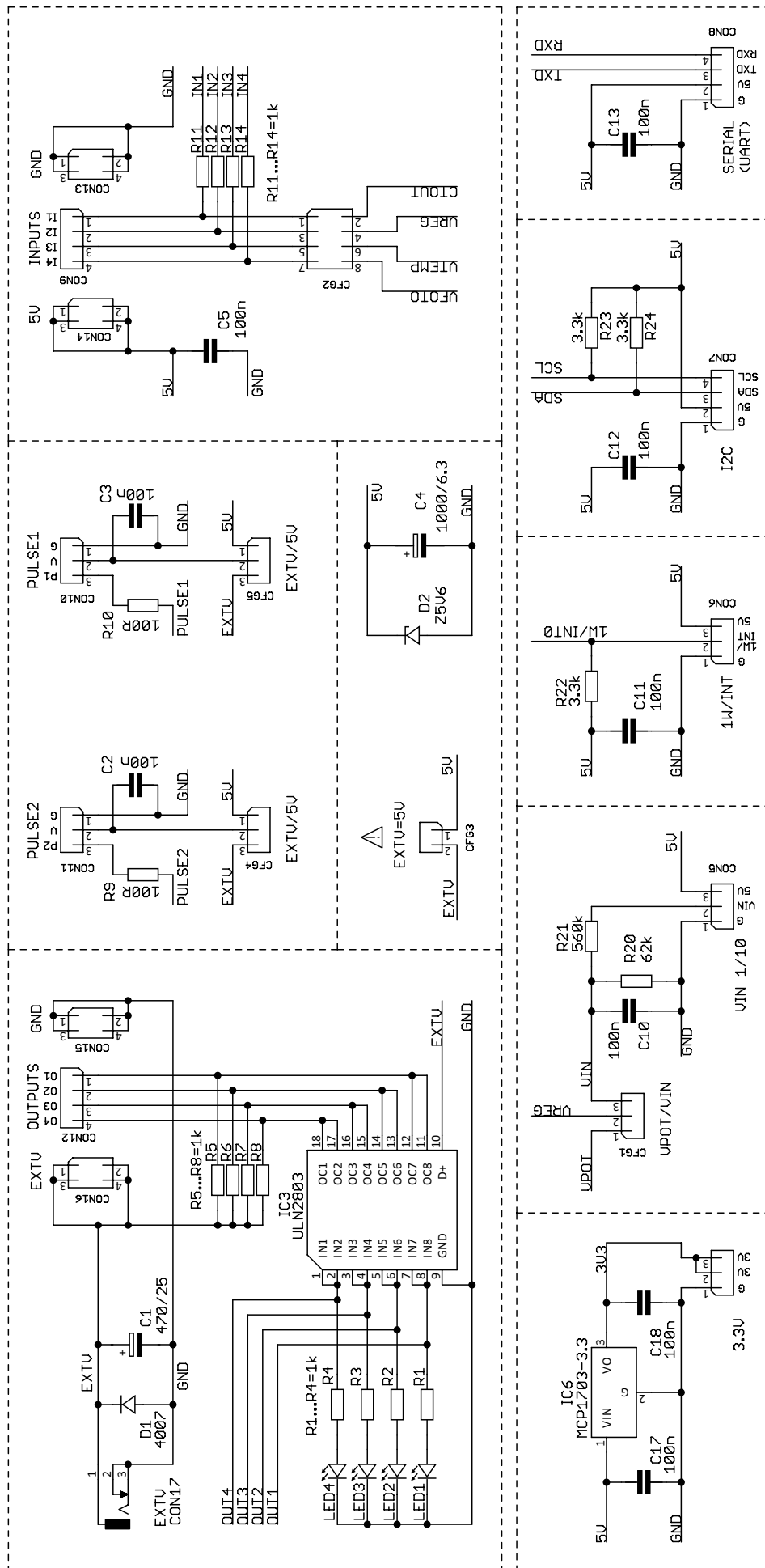
Podstawowe informacje:

- Złącze kompatybilne z płytką Arduino Uno.
- Podstawowe elementy interfejsu użytkownika (wyświetlacz 2x16, 4xprzycisk, 4xLED).
- 4 wyjścia ze stopniem mocy ULN2803 (max 0,5 A, max 24 V) i oddzielnym złączem zasilającym.
- 4 wejścia dla sygnałów analogowych i/lub cyfrowych.
- 2 wyjścia dla serwomechanizmów lub wyjścia sygnału PWM.
- Złącza interfejsów 1-Wire, I²C, UART.
- Wejście analogowe z dzielnikiem 1/10 (możliwość pomiaru napięcia do 50 V).
- Komponenty dodatkowe: sygnalizator dźwiękowy, fotorezystor, czujnik temperatury, potencjometr, czujnik dotykowy.
- Wyświetlacz sterowany interfejsem szeregowym UART (wymaga tylko 1 wyprowadzenia do komunikacji).

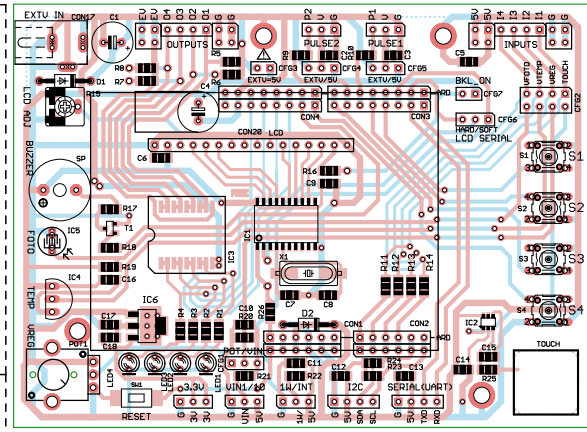
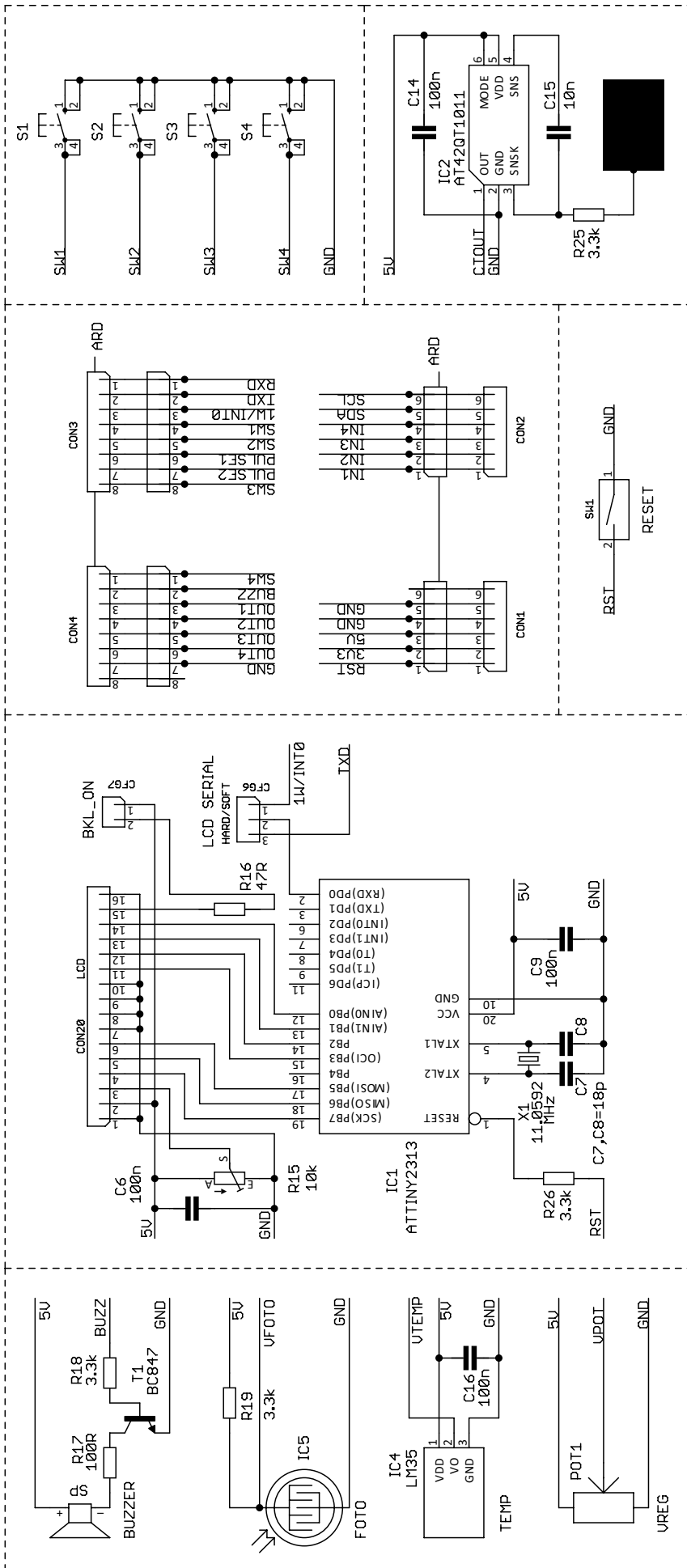
Projekty pokrewne na www.media.avt.pl:

AVT-1795	AVTduino Battery Shield (EP 3/2014)
AVT-1722	AVTduino miniLCD – miniaturowy panel operatora dla Arduino (EP 1/2013)
AVT-1686	AVTRelDuino Shield. Moduł wykonawczy dla Arduino (EP 8/2012)
AVT-5351	AVTduino RS. Moduł interfejsów szeregowych dla Arduino (EP 7/2012)
AVT-1666	AVTduino RELAY. Moduł przekaźników kompatybilny z Arduino (EP 3/2012)
AVT-1649	AVTduino SD. Moduł karty pamięci kompatybilny z Arduino (EP 11/2011)
AVT-1619	AVTduino Motor – driver silników dla Arduino (EP 9/2011)
AVT-1618	AVTduino JOY – manipulator dla Arduino (EP 6/2011)
AVT-1616	AVTduino LED. Wyświetlacz LED dla Arduino (EP 4/2011)
AVT-1615	AVTduino LCD. Wyświetlacz LCD dla Arduino (EP 3/2011)
AVT-5272	AVTduino – pomysł na AVR (EP 1/2011)

* Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętności lutownicza!
 Podstawową wersją zestawu jest wersja [B] nazywana potocznie KiTem (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.
 Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:
 • wersja [C] zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytce PCB)
 • wersja [A] płytka drukowana bez elementów i dokumentacja
 • wersja [A+] płytka drukowana z scalonym wymagającym zaprogramowania, posiadając następujące dodatkowe wersje:
 • wersja [A+] płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja
 • wersja [UK] zaprogramowany układ
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>



Rysunek 2. Schemat ideowy płytki edukacyjnej dla Arduino



Rysunek 3. Schemat montażowy płytki edukacyjnej dla Arduino

Arduino należy wybrać polecenie *Szkic* a następnie *Dołącz Bibliotekę* – na liście powinna być teraz dostępna nasza biblioteka. Po kliknięciu na nią, w kodzie programu pojawi się nowa linia o treści:

```
#include <AVT5636lib.h>
```

Aby móc korzystać z metod biblioteki musimy utworzyć obiekt klasy naszej biblioteki. Pomijając wyjaśnienia, musimy wpisać jeszcze przed sekcją *setup()* następującą linię

```
AVT5636 myBoard;
```

Nazwa obiektu może być inna niż *myBoard* ale wtedy, w pozostałej części programu, również należy używać tej innej nazwy. Ostatnią linią, którą musimy umieścić w kodzie, jest wywołanie metody *init()* w sekcji *setup()*, tak jak poniżej:

```
void setup() {
    myBoard.init();
}
```

Teraz płytki edukacyjna, wspomagana wydawaną biblioteką, jest gotowa do pracy.

W katalogu biblioteki umieszczony jest plik *AVT5636example.txt* z przykładowym kodem programu, prezentującym różne funkcje i możliwości płytki. Źródła programów oraz biblioteki dostępne są w materiałach dodatkowych do tego artykułu. W razie problemów z oprogramowaniem płytki, warto poświęcić odrobinę czasu na przeanalizowanie tych źródeł.

KS

Tabela 1. Opis złączy płytki edukacyjnej


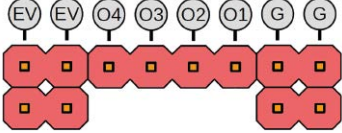
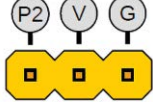
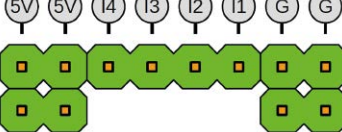
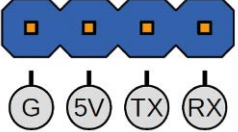
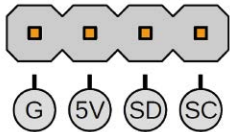
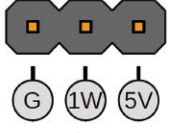
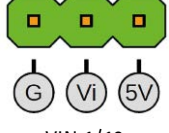
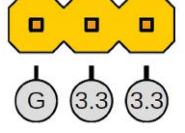
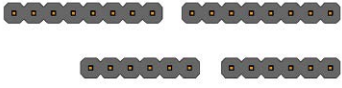
Złącza	Charakterystyka
<p>CON 17</p>  <p>EXTV</p>	<p>Gniazdo dla dodatkowego zasilania. Zalecany zakres napięcia 3...24 V. Zasilanie doprowadzone do złącza OUTPUTS i opcjonalnie do złączy PULSE. Przy doprowadzeniu 5 V możliwe zasilanie płytki edukacyjnej i płytki Arduino.</p>
<p>CON 12</p>  <p>OUTPUTS</p>	<p>4 wyjścia ze stopniem mocy z driverem ULN2803. Maksymalne obciążenie: 0,4 A dla przebiegu impulsowego lub 0,2 A dla stanu ciągłego, dla każdego z wyjść. Sterowanie: 5 V, stan aktywny na wyjściach O1...O4 – minus zasilania. Stan wejść sterujących połączony i obrazowany za pomocą LED1...LED4. Maksymalne napięcie: 24 V. Pozwala dołączyć np. silnik krokowy, przekaźniki, itp.</p>
<p>CON 10, CON 11</p>  <p>PULSE1, PULSE2</p>	<p>2 wyjścia dla sygnałów impulsowych. Połączone z wyjściami OC0A i OC0B płytki Arduino – umożliwiają sprzętowe generowanie przebiegów PWM. Złącza kompatybilne ze złączem serwomechanizmu. Na złączach zasilanie 5 V lub z zewnętrznego zasilacza (EXTV). Pozwala dołączyć np. serwomechanizm lub tranzystor MOSFET.</p>
<p>CON 9</p>  <p>INPUTS</p>	<p>4 wejścia dla sygnałów analogowych lub cyfrowych. Dopuszczalny zakres napięć: 0...5 V. Połączone z wejściami przetwornika ADC na płytce Arduino. Możliwe dołączenie komponentów dodatkowych z płytki edukacyjnej: czujnika światła, czujnika temperatury, potencjometru i włącznika dotykowego.</p>
<p>CON 8</p>  <p>SERIAL (UART)</p>	<p>Złącze interfejsu szeregowego UART. Zakres napięć dla sygnałów: 0...5 V. TX – sygnał wychodzący, RX – sygnał doprowadzany do Arduino. Sygnał TX może być wykorzystany do sterowania wyświetlaczem na płytce edukacyjnej. Pozwala dołączyć np. moduł interfejsu RS485 lub moduł Bluetooth</p>
<p>CON 7</p>  <p>I2C</p>	<p>Złącze interfejsu I2C. Zakres napięć dla sygnałów: 0...5 V. Wyposażone w rezystory podciągające (pull-up). Pozwala dołączyć różnego rodzaju czujniki np. ciśnienia i układy i/o np. potencjometry cyfrowe.</p>
<p>CON 6</p>  <p>1W / INTO</p>	<p>Może pełnić rolę złącza interfejsu 1-Wire. Może pełnić rolę przerywania sprzętowego – połączone z sygnałem INT0 płytki Arduino. Zakres napięć dla sygnałów: 0...5 V. Wyposażone w rezystor podciągający (pull-up). Pozwala dołączyć np. czujnik temperatury DS18B20, odbiornik podczerwieni TSOP1736, itp.</p>
<p>CON 5</p>  <p>VIN 1/10</p>	<p>Wejście dla sygnału analogowego z wbudowanym dzielnikiem napięcia 1/10. Zakres napięcia wejściowego: 0...50 V. Pozwala dołączyć w bezpieczny sposób sygnał o napięciu większym niż 5 V, np. z zasilacza regulowanego.</p>
<p>CON 4</p>  <p>3,3 V</p>	<p>Wyjście napięcia 3,3 V z niezależnego stabilizatora. Maksymalne obciążenie 250 mA.</p>
<p>CON1...CON4</p> 	<p>Szpilki dołączone równolegle i bezpośrednio do złączy płytki Arduino, wyprowadzone po stronie płytki edukacyjnej.</p>

Tabela 2. Opis szpilek konfiguracyjnych płytki edukacyjnej


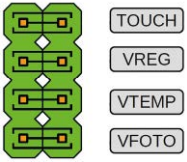
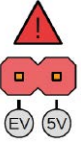
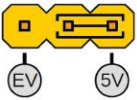

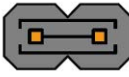
Szpilki konfiguracyjne	Funkcja
<p>CFG1</p>  <p>VPOT/VIN</p>	Pozwala wybrać czy wejście I2 złącza INPUTS będzie połączone z potencjometrem (zwora w poz. VPOT) czy z wejściem analogowym z dzielnikiem napięcia (zwora w poz. VIN).
<p>CFG2</p>  <p>TOUCH VREG VTEMP VFOTO</p>	Brak zworek – sygnały doprowadzone będą ze złącza INPUTS . Gdy zworki założone – do wejść złącza INPUTS dołączone są sygnały z komponentów dodatkowych: czujnika światła, czujnika temperatury, potencjometru i włącznika dotykowego.
<p>CFG 3</p>  <p>EV 5V EXTV = 5 V</p>	Pozwala dołączyć napięcie ze złącza zasilania zewnętrznego EXTV do szyny zasilającej 5 V płytki edukacyjnej i płytki Arduino. W ten sposób płytki będą zasilane ze złącza EXTV (bez potrzeby zasilania z USB). Uwaga – dołączone napięcie musi mieć wtedy wartość 5 V DC.
<p>CFG 4, CFG 5</p>  <p>EV 5V EXTV/5 V</p>	Pozwala wybrać czy do wyjść impulsowych PULSE1 i PULSE2 , do szpilek zasilających, oznaczonych V, ma być doprowadzone napięcie 5V (zwora w poz. 5 V) czy napięcie z gniazda zasilania EXTV (zwora w poz. EXTV).
<p>CFG 6</p>  <p>HARD/SOFT LCD SERIAL</p>	Pozwala wybrać czy wyświetlacz na płytce edukacyjnej będzie sterowany poprzez sprzętowy interfejs szeregowy UART (zwora w poz. HARD) czy poprzez programowy port szeregowy, który musi być przypisany do wyprowadzenia D2 (zwora w poz. SOFT).
<p>CFG 7</p>  <p>BKL ON</p>	Pozwala włączyć podświetlenie wyświetlacza zamontowanego w płytce edukacyjnej (zwora założona).

Tabela 3. Opis metod dedykowanej biblioteki

Metody biblioteki	Krótką charakterystyka
<code>void init();</code>	Konfiguruje funkcje portów. Przygotowuje sprzętowy port szeregowy UART do sterowania wyświetlaczem. musi być umieszczona wewnątrz sekcji <code>setup()</code> , programu w Arduino.
<code>void outOn(int num);</code> <code>void outOff(int num);</code> <code>void outPulse(int num, int tIme);</code>	Włącza stan aktywny na wyprowadzeniu złącza OUTPUTS . Wyłącza stan aktywny na wyprowadzeniu złącza OUTPUTS . Generuje impuls o zadanym czasie, stanu aktywnego na wyprowadzeniu złącza OUTPUTS . parametr <i>num</i> to numer wypr. (1, 2, 3 lub 4), <i>tIme</i> – czas w milisekundach.
<code>bool outGet(int num);</code>	Zwraca stan jednego wyprowadzenia złącza OUTPUTS . 1 – wypr. włączone, 0 – wypr. wyłączone.
<code>int outputsGet();</code> <code>void outputsSet(int state);</code>	Zwraca stan wszystkich wypr. złącza OUTPUTS . Ustawia stan wszystkich wypr. złącza OUTPUTS . wartość 0x00 – wyprowadzenia wyłączone, 0x0F – włączone.
<code>void ledOn(int num);</code> <code>void ledOff(int num);</code> <code>void ledPulse(int num, int tIme);</code>	Włącza diodę LED o zadanym numerze. Wyłącza diodę LED o zadanym numerze. Włącza diodę LED o zadanym numerze na określony czas. Parametr <i>num</i> to numer diody LED (1, 2, 3 lub 4), <i>tIme</i> – czas w milisekundach. Diody LED połączone są z wypr. OUTPUTS , zmiana stanu diody przekłada się na zmianę stanu na wypr. OUTPUTS .
<code>void pulse1Set(int val);</code> <code>void pulse2Set(int val);</code>	Generuje na wypr. PULSE1 sygnał PWM o zadanym wypełnieniu. Generuje na wypr. PULSE2 sygnał PWM o zadanym wypełnieniu. <i>val</i> – wartość z przedziału 0...255 odpowiada wypełnieniu 0..100%.
<code>void buzzerOn();</code> <code>void buzzerOff();</code> <code>void buzzerPulse(int tIme);</code>	Włącza sygnalizator dźwiękowy. Wyłącza sygnalizator dźwiękowy. Włącza sygnalizator dźwiękowy na określony czas. <i>tIme</i> – czas w milisekundach.

Tabela 3. cd.

Metody biblioteki	Krótką charakterystyka
<code>int getButtons();</code>	Zwraca stan przycisków S1...S4, możliwe wartości to 0, 1, 2, 3 lub 4.
<code>int getInputs();</code> <code>bool getInput(int num);</code>	Zwraca stan wszystkich wypr. złącza INPUTS . Wartość 0x00 – wypr. w stanie niskim, 0x0F – wypr. w stanie wysokim.. Zwraca stan jednego wypr. złącza INPUTS . Parametr <i>num</i> to numer wypr. (1, 2, 3 lub 4) złącza INPUTS .
<code>int getInputAdc(int num);</code> <code>int getInputVoltage(int num);</code> <code>int getInputPercent(int num);</code>	Zwraca wartość odczytaną przez przetw. ADC z danego wejścia. Zwraca wartość w mV, odczytaną przez przetw. ADC z danego wejścia. Zwraca wartość w %, odczytaną przez przetw. ADC z danego wejścia. Parametr <i>num</i> to numer wejścia. (1, 2, 3 lub 4) złącza INPUTS . zakres wartości wynosi: 0...1023; 0...5000mV; 0...100%.
<code>int getTouch();</code> <code>int getVreg();</code> <code>int getLight();</code> <code>int getTemp();</code> <code>int getVin0();</code>	Zwraca stan wyjścia czujnika dotykowego, 1 – aktywne, 0 – nieaktywne. Zwraca wartość w mV, odczytaną z położenia potencjometru VREG . Zwraca wartość w %, odczytaną z czujnika światła FOTO . Zwraca wartość w stopniach C, odczytaną z czujnika temperatury TEMP . Zwraca wartość w mV, odczytaną z wejścia pomiaru napięcia VIN . Odczyt z czujników możliwy jest po odpowiednim ustawieniu zwerek na szpilkach konfiguracyjnych CFG1 i CFG2 .
<code>void lcdPrint(String str);</code> <code>void lcdPosition(int y, int x);</code> <code>void lcdClear();</code> <code>void lcdSpecChar(int index, char *src);</code> <code>void lcdMessage(int y, int x, String str);</code>	Wysyła treść na wyświetlacz LCD. Ustawia współrzędne dla wysyłania treści na wyświetlacz LCD. Czyści treść wyświetlacza i ustawia współrzędne na początek ekranu. Pozwala skonfigurować własny znak w pamięci wyświetlacza. Sterownik ustawia znak stopni w baku pamięci o numerze 1, można go wyświetlić umieszczając w treści do wyświetlenia ciąg “\1”. Wysyła treść na wyświetlacz lcd, na podane współrzędne. <i>y</i> – numer wiersza 1...2; <i>x</i> – numer znaku 1...16; <i>str</i> – zmienna String;

REKLAMA

Klub Aplikantów Próbek

to inicjatywa redakcji Elektroniki Praktycznej. W kontaktach z firmami redakcja często otrzymuje do przetestowania próbki podzespołów, modułów, a nawet całych urządzeń elektronicznych. Są to zwykle najnowsze typy/modele produktów na rynku. Z chęcią podzielenia się z Czytelnikami tymi próbkami zrodziła się inicjatywa pod nazwą Klub Aplikantów Próbek. Członkiem KAP staje się każdy, kto zgłosi chęć przetestowania próbki. Wykaz i krótki opis próbek, którymi dysponuje redakcja EP, można znaleźć poniżej (www.ep.com.pl/KAP). Wystarczy wybrać rodzaj próbek i zwrócić się majłem (na adres: Szef Pracowni Konstrukcyjnej grzegorz.becker@ep.com.pl) z prośbą o przesłanie bezpłatnych próbek, podając ich nazwę i adres wysyłki. Warto dopisać jaki jest plan zastosowania tych próbek. Nie jest to konieczne, ale może mieć znaczenie przy podziale próbek w przypadku większej liczby zgłoszeń. Mile widziane, choć nieobowiązkowe, jest też przysłanie do redakcji EP opisu wykonanej aplikacji próbek, oczywiście po jej wykonaniu z zastosowaniem otrzymanej próbki. Autorom przysłanych opisów przyznamy punkty, które będą im dawały pierwszeństwo przy ubieganiu się o kolejne próbki. Najciekawsze opisy aplikacji opublikujemy na forum ep.com.pl lub na łamach Elektroniki Praktycznej. Dla pełnej jasności jeszcze raz podkreślamy, że próbki przekazujemy bezpłatnie i nie trzeba ich zwracać do redakcji.



www.ep.com.pl/kap