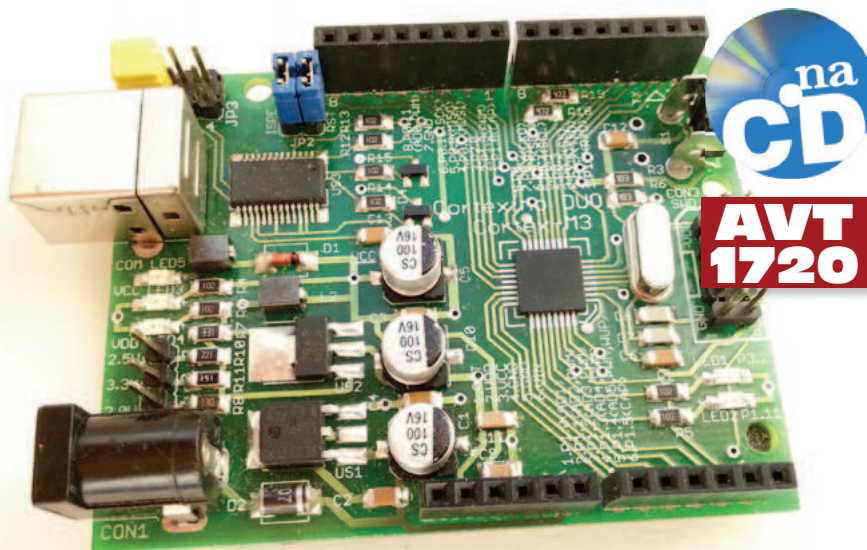


# Cortexino Duo – płyta ewaluacyjna z LPC1343

Płytką łączy moc mikrokontrolera 32-bitowego z elastycznością zastosowań Arduino. Przyda się wszędzie tam, gdzie moc 8-bitowego rdzenia AVR jest niewystarczająca. Mikrokontroler zamontowany na płytce ma zaprogramowany bootloader oraz sterowniki USB HID i MSD.



Schemat płytki pokazano na rysunku 1. Zasilanie płytki może być pobierane z portu USB, wtedy prąd nie powinien przekraczać 100 mA, a napięcie VCC będzie miało wartość ok. 4,5 V. Jeżeli przewidujemy większy pobór prądu np. przez peryferia dołączone do płytki, to należy zasilić układ z zewnętrznego zasilacza 7...12 V DC. Wtedy napięcie VCC jest dostarczane przez stabilizator US1 i wynosi 5 V. Stabilizator można obciążyć prądem do 0,5 A, co jest wystarczające w większości wypadków.

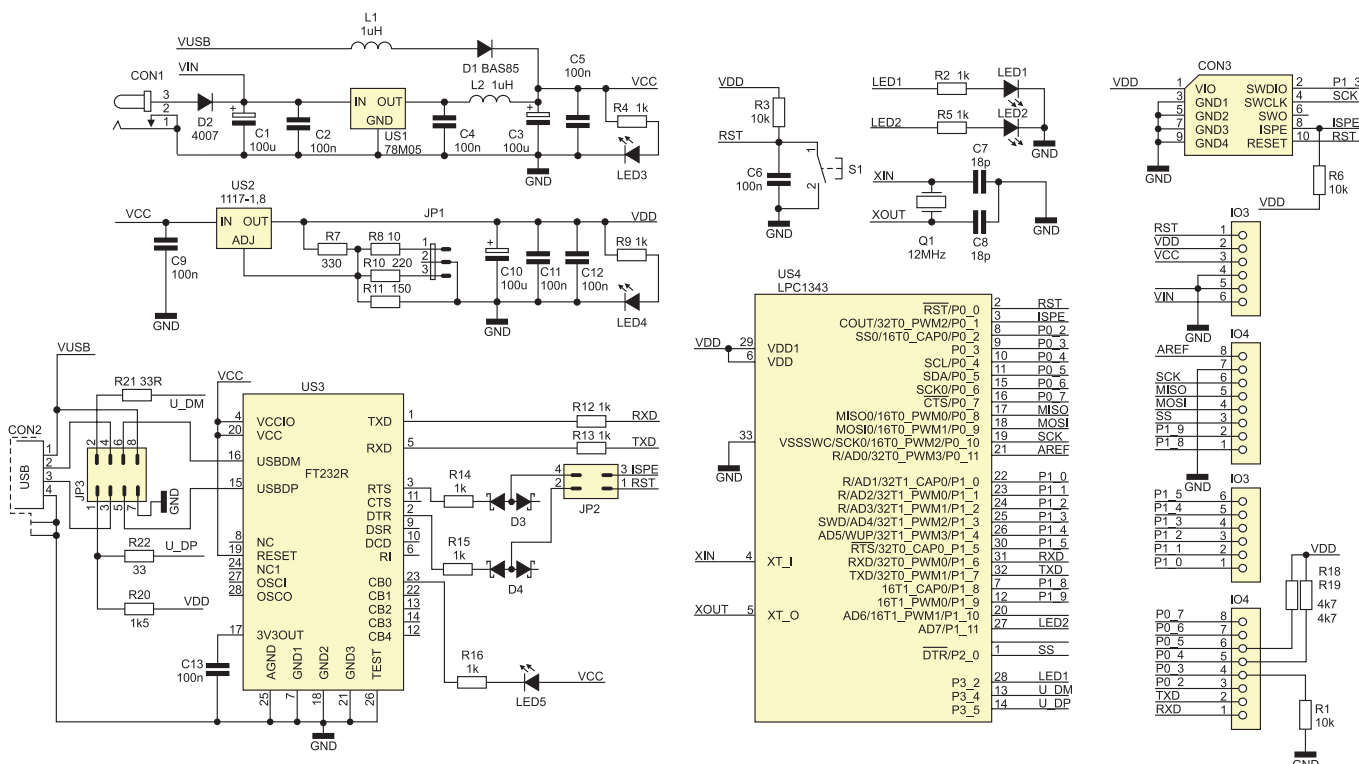
Mikrokontroler jest zasilany napięciem 2,0; 2,5 lub 3,3 V wybieranym za pomocą zworki JP1. Diody LED3 i LED4 sygnalizują obecność napięć zasilających. Układ US3 to konwerter USB/UART, który pełni dwie funkcje: pozwala na programowanie pamięci mikrokontrolera w trybie ISP oraz na transmisję danych do komputera PC za pośrednictwem interfejsu szeregowego, co może być wyko-

rzystane jako prosty debugger. Złącze JP2, po założeniu dwóch jumperów, pozwala na dołączenie sygnałów RESET i ISP ENABLE do wyprowadzeń układu FT232, a to zapewnia pełną automatyzację procesu programowania. Pracę interfejsu USB sygnalizuje dioda świecąca LED5. Przycisk S1 służy do zresetowania mikrokontrolera, Q1 jest źródłem sygnału taktującego, diody LED1 i LED2 pełnią rolę sygnalizacyjną, mogą być wykorzystane w dowolny sposób. Porty mikrokontrolera doprowadzono do złącz IO2, IO3 i IO4.

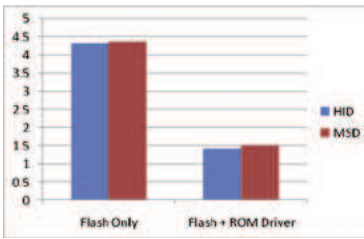
Złącze CON3 umożliwia dołączenie zewnętrznego programatora/debuggera z interfejsem SWD. Złącze JP3 służy do wyboru urządzenia USB: konwerter FT232 lub mi-

krokontroler. Rozmieszczenie złącz jest kompatybilne z Arduino Duemilano Board, co pozwala na zastosowanie modułów rozszerzających (shields) dla Arduino. Doprowadzenia PORT0\_4 i PORT0\_5 są także szyną I<sup>2</sup>C i pracują jako wyjścia open drain, więc zostały wyposażone w zewnętrzne rezystory podciągające.

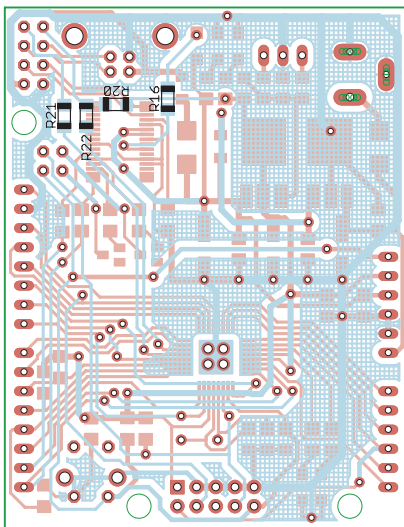
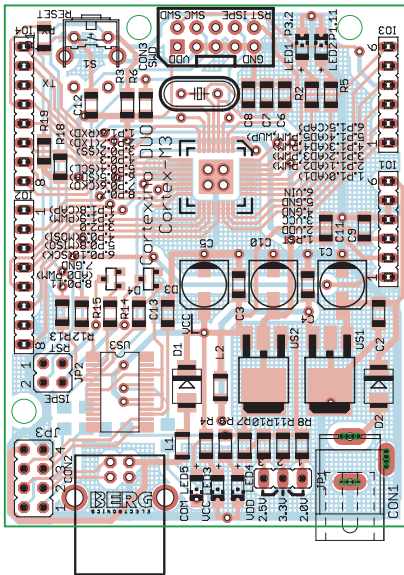
Układ LPC1343 ma większość peryferiów charakterystycznych dla „małych” mikrokontrolerów: timery, przetwornik A/C, interfejsy szeregowy, kontroler przerw itd. Wyróżniającym komponentem jest interfejs USB. Na uwagę „on-chip drivers”, to jest oprogramowanie zapisane w pamięci ROM na etapie produkcji. Podobnym elementem



Rysunek 1. Schemat ideowy płytki CortexinoDuo



Rysunek 2. Porównanie rozmiaru kodu wynikowego.



Rysunek 3. Schemat montażowy płytki CortexinoDuo

tego typu są bootloadery zwykle służące do zapisywania lub odczytu pamięci programu. LPC1343 oprócz bootloadera ma w pamięci grupę funkcji, które tworzą rodzaj interfejsu programowania aplikacji – API i pozwalają realizować złożone funkcje interfejsu USB – emulować urządzenie USB typu HID lub MSD.

Architektura USB składa się z urządzenia nadrzędnego *hosta* i jednego lub wielu urządzeń *device*. Każde urządzenie komunikuje się z kontrolerem za pośrednictwem kanałów logicznych – *pipes*, których może być do 32. Dostęp do kanałów komunikacyjnych

jest możliwy dzięki buforom *endpoints*, ponieważ są one zakończeniami tych kanałów. Rodzaj urządzenia określa *device class*. Może to być np.: HID, MSD itp. Szczegółowe informacje o urządzeniu, m.in.: nazwa, numery identyfikacyjne VID i PID, zawarte są w bloku *device descriptor*. Proces wykrywania i rozpoznawania jest nazywany *enumeracją*. W jego trakcie urządzenie otrzymuje adres, odczytywane są informacje z bloku deskryptora oraz przydzielane sterowniki.

Interfejs API w pamięci mikrokontrolera LPC1343 umożliwia: uruchomienie części sprzętowej interfejsu, nawiązanie połączenia z kontrolerem oraz otwarcie 2 kanałów komunikacyjnych (wejściowego i wyjściowego). Jest on dostępny jako zespół 4 funkcji, które muszą być wywołane w odpowiedniej kolejności i z odpowiednimi parametrami. W uproszczeniu, na przykładzie konfiguracji urządzenia HID w trybie *device*, przebiega to w sposób następujący:

1. Uruchomienie funkcji inicjującej sprzęt:

```
(*rom)->pUSB->init_clk_pins();
```

2. Ustawienie parametrów i zainicjowanie sterownika:

```
HidDevInfo.idVendor = USB_VEN-DOR_ID;
HidDevInfo.idProduct = USB_PROD-ID;
HidDevInfo.bcdDevice = USB_DE-VICE;
HidDevInfo.StrDescPtr = (uint32_t)&USB_StringDescrip-tor[0];
HidDevInfo.InReportCount = 1;
HidDevInfo.OutReportCount = 1;
HidDevInfo.SampleInterval = 0x20;
HidDevInfo.InReport = GetInRe-port;
HidDevInfo.OutReport = SetOutRe-port;
DeviceInfo.DevType = USB_DEVICE-CLASS_HUMAN_INTERFACE;
DeviceInfo.DevDetailPtr = (uint32_t)&HidDevInfo;
(*rom)->pUSB->init(&DeviceInfo);
```

3. Ustawienie funkcji dla przerwania oraz dla odbioru i wysyłania danych:

```
USB_IRQHandler(void) {(*rom)->pUSB->isr();}
void GetInReport(uint8_t src[], uint32_t length) {src[0] = LPC_GPIO0->DATA; // Switch state}
void SetOutReport(uint8_t dst[], uint32_t length) {LPC_GPIO2->DATA = dst[0]; // Set LED state}
```

4. Otwarcie połączenia

```
(*rom)->pUSB->connect(TRUE);
/* USB Connect
```

API w pamięci mikrokontrolera umożliwia łatwe tworzenie aplikacji, co przekła-

**Wykaz elementów**

**Rezystory:** (SMD 0805)

- R2, R4, R5, R9, R12...R16: 1 kΩ
- R1, R3, R6: 10 kΩ
- R7: 330 Ω
- R8: 33 Ω
- R10: 220 Ω
- R11: 150 Ω
- R18, R19: 4,7 kΩ
- R21, R22: 33 Ω
- R20: 1,5 kΩ

**Kondensatory:** (SMD 0805)

- C1, C5, C10: 100 μF / 16 V (SMD „C”)
- C2, C4...C6, C9, C11...C13: 100 nF
- C7, C8: 18 pF

**Półprzewodniki:**

- D1: BAS85
- D2: 1N4007 (SMD)
- D3, D4: BAR43
- LED1...LED5: LED (SMD)
- U1: 78M05
- U2: LM1117-1.8
- U3: FT232R
- U4: LPC1343 (HQFN33)

**Inne:**

- L1, L2: 1...10 μH (dławik SMD)
- S1: przycisk kątowy
- Q1: kwarc 12 MHz
- JP1: goldpin 1×3+zworka
- JP2: goldpin 2×2+zworki
- JP3: goldpin 2×4+zworki
- IO1, IO3: gniazdo goldpin 1×6
- IO2, IO4: gniazdo goldpin 1×8
- CON1: gniazdo kątowe DC2.1/5.5
- CON2: gniazdo kątowe USB B
- CON3: goldpin 2×5

**W ofercie AVT\***

- AVT-1720 A AVT-1720 B
- AVT-1720 C

**Dodatkowe materiały na CD/FTP:**

<ftp://ep.com.pl>, user: 13621, pass: 175brjf7

- 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-5362 AVTDuinoMEGA (EP 9/2012)
- AVT-1707 DIO-Expander - moduł ekspandera portów dla AVTDuino (EP 9/2012)
- AVT-1686 AVTReDuino Shield - moduł wykonawczy dla Arduino (EP 8/2012)
- AVT-5351 AVTDuino RS - moduł interfejsów szeregowych dla Arduino (EP 7/2012)
- AVT-1677 AVTDuino PWM (EP 6/2012)

\* Uwaga: Zestawy AVT mogą występować w następujących wersjach: AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych. AVT xxxx A płytką drukowaną PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych. AVT xxxx A+ płytką drukowaną i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych. AVT xxxx B płytką drukowaną (lub płytki) oraz komplet elementów wymieniony w załączniku pdf to nie 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>

da się na oszczędność czasu i zmniejszenie wielkości programu. Na **rysunku 2** pokazano ilustrację z dokumentacji firmy NXP z porównaniem wielkości kodu wynikowego w kB typowego programu i wykonanego z użyciem API.

Schemat montażowy CortexinoDuo pokazano na **rysunku 3**. Jego montaż jest typowy, chociaż wymaga wprawy i uwagi, ponieważ płytką składa się głównie z elementów SMD.