

KOŁO NAUKOWE ROBOTYKÓW  
„KONAR”



KONAR

KOŁO NAUKOWE ROBOTYKÓW

## Interfejsy OSBDM i TBDML

Robert Kuczaj

Wrocław, 11 marca 2008

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Montaż układu</b>	<b>2</b>
<b>3</b>	<b>Uruchomienie układu</b>	<b>3</b>
3.1	Programowanie układu . . . . .	7
3.1.1	Aplikacja USBICP . . . . .	7
3.1.2	Loader oparty o libusb . . . . .	7
3.2	Interfejs Open Source BDM . . . . .	8
3.3	Interfejs Turbo BDM Light . . . . .	8
3.4	Zworka zasilania . . . . .	8
3.5	Zmiana firmware'u . . . . .	8
<b>4</b>	<b>Schemat układu i projekt PCB</b>	<b>9</b>



Rysunek 1: Interfejs TBDML i OSBDM

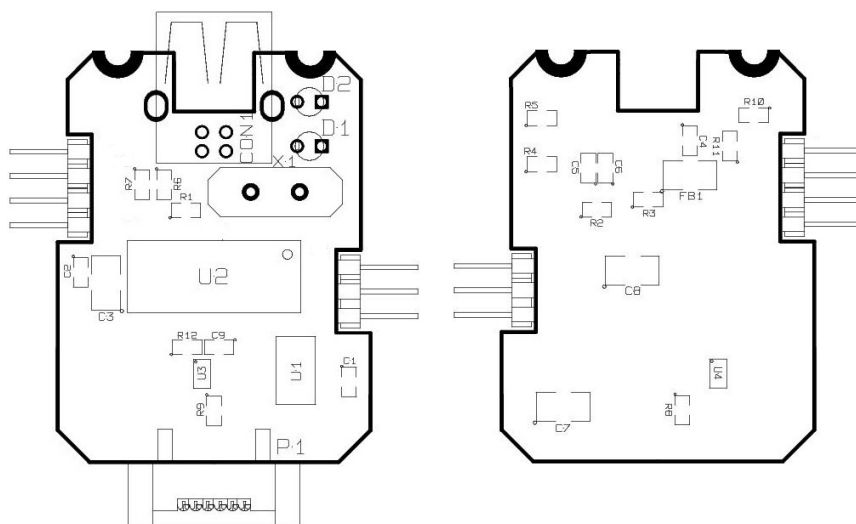
## 1 Wstęp

Opisany w dokumencie interfejs BDM (Background Debug Module) został stworzony w oparciu o projekty interfejsów OsBDM [1] i TBDML[2] oraz informacje dostępne na forum dyskusyjnym [3, 4] firmy Freescale [5]. Pozwala on na uruchomienie i testowanie procesorów z rodziny HCS(R)08 lub HC12 w zależności od wersji użytego oprogramowania. Dzięki podobieństwu oryginalnych projektów oraz dostępności informacji i wsparciu ze strony producenta możliwe było przeniesienie obydwu interfejsów na jedną platformę sprzętową opartą o układ HC908JB16 przy niewielkich zmianach firmware'u urządzeń. Zachowana została kompatybilność z ich oryginalnymi wersjami dzięki czemu są one oficjalnie wspierane przez środowisko CodeWarrior. Widok interfejsu przedstawiono na rysunku 1.

Układ posiada możliwość pracy z procesorami z rodziny HCS(R)08 lub HC12(X) w zależności od wykorzystanego firmware'u. Jest wspierany przez środowisko CodeWarrior, narzędzie przygotowane przez producenta układów i dostępne za darmo przy pewnych ograniczeniach wielkości kodu. Zaletą układu jest możliwość zasilania układu docelowego napięciem 3.3V lub 5V bezpośrednio z magistrali USB (pobór prądu nie może być większy niż 100mA) oraz może pracować z układem zasilanym dowolnym napięciem w zakresie 1.2V-5V pod warunkiem zasilania interfejsu z urządzenia, do którego jest podłączony.

## 2 Montaż układu

Płytką drukowaną wykonaną dla zaprojektowanego interfejsu posiada opisy elementów, których położenie dodatkowo umieszczono na rysunku 2. Montaż układu należy rozpocząć od układów U2, U3, U4. Następnie należy przylutować pozostałe elementy SMD (biegun dodatni kondensatorów tantalowych oznaczony jest kółkiem przy odpowiednim padzie) w kolejności od najmniejszych do największych. Na końcu montujemy goldpiny, które będą służyć jako zworki (J1, J2, J3, J4, P2) oraz złącze P1 (rysunek 3). Złącz stanowiących zworki Z1 i Z2 można nie montować i zewrzeć padu po obydwu stronach płytki srebrząnką. Zworki te zostały przewidziane dla bardziej wymagających magistral USB zdarzających się w niektórych laptopach jednak w praktyce nie ma

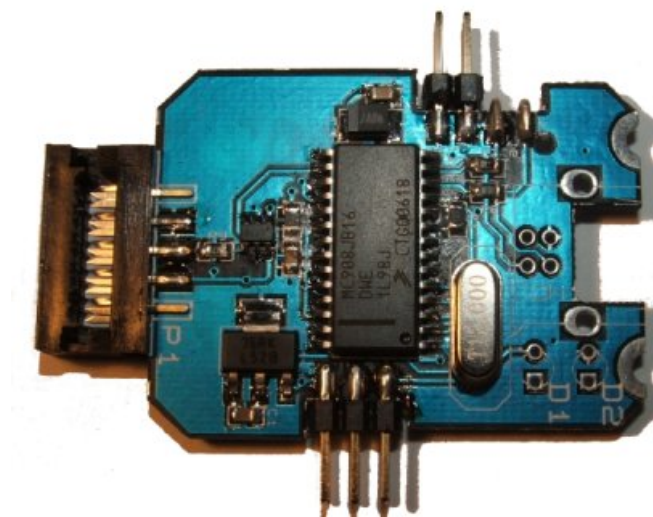


Rysunek 2: Rozmieszczenie elementów na płycie drukowanej

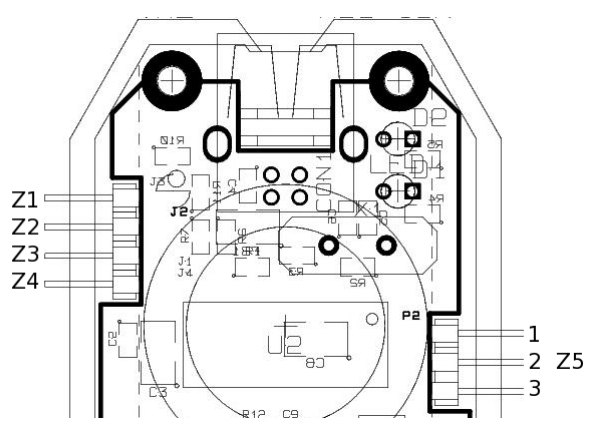
potrzeby ich stosowania. Opis zwojek przedstawiono na rysunku 4. Następnie należy taśmę kablową rozerwać w taki sposób aby pozostawić tylko 6 przewodów i zacisnąć ją w złączu P1 (na jego środku a nie przy brzegu) a na jej końcu złącze ICD6 zwracając uwagę na położenie pierwszego pinu. Widok poprawnie zamontowanej taśmy przedstawiono na rysunku 5. Kolejnym etapem jest wlutowanie kabla USB, który należy obciąć z jednej strony pozostawiając złącze męskie typu A do podłączenia programatora do komputera. Warto wykorzystać ekran kabla do jego zamocowania w płycie. W tym celu należy ekran rozdzielić na dwie części usuwając folie aluminiową (nie da się jej lutować), spleść druty i wlutować je w otwory montażowe złącza USB. Dopiero tak przytwierdzony kabel lutujemy do płytki. Sposób montażu kabla pokazano na rysunku 6. Przed wlutowaniem kabla należy umieścić na nim gumowy przepust. Ostatnim etapem montażu jest umieszczenie płytki w obudowie i przylutowanie diod na odpowiedniej wysokości. W tym celu należy wyłamać lub wyciąć z obudowy fragment wewnętrznej ścianki jak pokazano na rysunku 7 oraz wywiercić otwór na dodatkową diodę w odległości około 3 mm od już istniejącego. Następnie umieszczamy nóżki diod o otworach w płycie, umieszczamy ją w górnej części obudowy i lutujemy diody na rządnej wysokości. Następnie obcinamy wystające nóżki i zamykamy obudowę umieszczając przepust w jej ścianie. Programator przez zamknięciem obudowy pokazano na rysunku 8. Skręcając obudowę należy jeszcze wypilować miejsce na zwoyki oraz rozerwać na fragmencie taśmę kablową aby ominąć śrubę montażową co pokazano na rysunku 9.

### 3 Uruchomienie układu

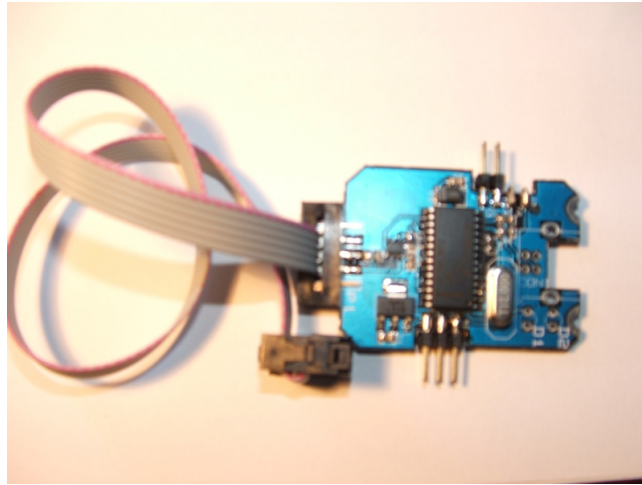
Dołączona do dokumenty paczka instalacyjna **osbdm\_tbdml.zip** zawiera niezbędne oprogramowanie, pozwalające uruchomić i zainstalować wybrany programator w systemie operacyjnym Windows. Dodatkowo dołączone zostały sterowniki dla obydwu programatorów. Aby urządzenie pracowało poprawnie



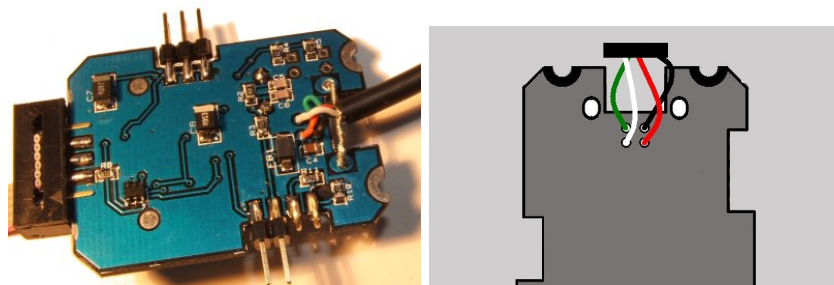
Rysunek 3: Płytką z zamontowanymi złączami



Rysunek 4: Opis zwerek interfejsu (widok z góry)



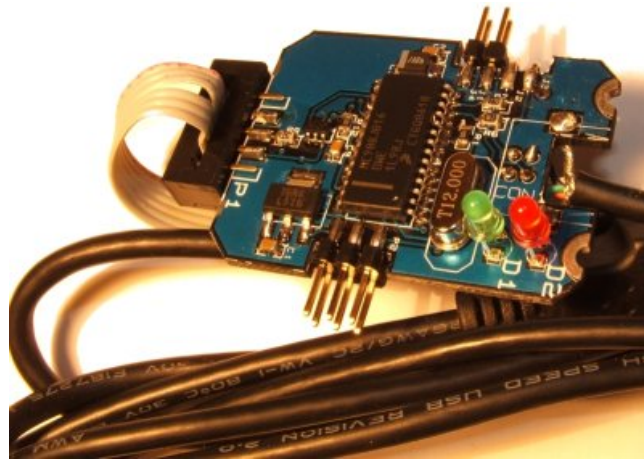
Rysunek 5: Płytką z zamontowaną taśmą kablową



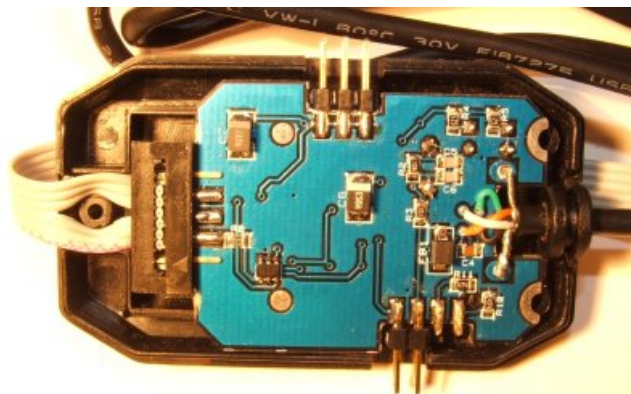
Rysunek 6: Sposób montażu kabla USB (widok z dołu).



Rysunek 7: Fragment, który należy wylamać z obudowy



Rysunek 8: Programator przed zamknięciem w obudowie



Rysunek 9: Montaż taśmy kablowej w obudowie oraz wycięcia pod zworki.

niezbędne jest wgranie odpowiedniego firmware'u a następnie jego instalacji a w systemie.

### 3.1 Programowanie układu

Wykorzystany w układzie kontroler firmy Freescale z rodziny JB16 posiada wbudowany w procesie produkcyjnym bootloader, dzięki któremu możliwe jest jego programowanie przez magistralę USB, bez konieczności użycia dodatkowych urządzeń pośredniczących. Tryb ten nazywa się ICP (In Circuit Programing) i został dokładnie opisany w nocie aplikacyjnej [6].

Aby zaprogramować zmontowany układ należy włożyć zworki Z1 i Z2 (jeśli zdecydowano się z nich korzystać) i podłączyć go do komputera i zainstalować odpowiednie sterowniki. Do wyboru są dwa możliwe sposoby programowania:

- Przy użyciu aplikacji USBICP dostarczonej przez producenta układu, który niestety nie uruchamia się na każdym komputerze
- Przy użyciu konsolowego loadera opartego o libusb stworzonego dla Turbo DMB Colf Fire.

Paczka instalacyjna programatorów zawiera obydwie aplikacje a wybór zależy od użytkownika.

#### 3.1.1 Aplikacja USBICP

System powinien wykryć nowe urządzenie USB i poprosić o sterowniki do niego. Należy wybrać instalację sterowników z określonej lokalizacji i jako ścieżkę do sterowników wskazać folder **osbdm\_tbdml\icp\_loader**. System powinien zainstalować urządzenie **Motorola ICP JB16/JG16**. Po zainstalowaniu urządzenia należy uruchomić aplikację USBICP znajdującą się w katalogu ze sterownikami. Program po uruchomieniu zapyta o odpowiednią mapę pamięci dla procesora. W katalogu znajduje się tylko jedna mapa dla procesora HC908JB16 (**jb16icp\_me.imp**) i to ją należy wybrać. Następnie klikami przycisk „Select File...” i wybieramy odpowiedni firmware `osbdm_firmware\opensourcebdm_jb16.s19` jeśli chcemy korzystać z OsBDM lub `tbdml_firmware\turbobdmllight_jb16.s19` jeśli chcemy korzystać z TBDML. Następnie po naciśnięciu przycisku Program nastąpi zaprogramowanie procesora.

#### 3.1.2 Loader oparty o libusb

Chcąc korzystać z loadera konsolowego (np. jeśli aplikacja USBICP nie uruchamia się na komputerze) należy dla podłączonego kontrolera zainstalować sterowniki z katalogu **osbdm\_tbdml\libusb\_loader**. W przypadku problemów z aplikacją USBICP należy wcześniej odinstalować urządzenie za pomocą Menadżera Urządzeń a następnie zainstalować nowe urządzenie USB z wybranej lokalizacji podając folder **osbdm\_tbdml\libusb\_loader** jako lokalizację sterowników. Po zainstalowaniu urządzenia możemy je zaprogramować używając programu **tblcf\_bt** z argumentem -B oraz ścieżka do odpowiedniego pliku. Przykładowe wywołanie programu dla programowania TBDML wygląda następująco:



```
>tblcf_bt.exe -B ..\tbdml_firmware\turbobdmlight_jb16.s19
```

Po zakończeniu programowania należy odłączyć urządzenie od komputera i włożyć zworkę Z3. Zworki Z1 i Z2 można pozostawić lub usunąć (jeśli zdecydowano się na ich używanie). Zaprogramowany interfejs jest gotowy do pracy.

### 3.2 Interfejs Open Source BDM

Po podłączeniu interfejsu do komputera po jego wcześniejszym zaprogramowaniu jako OsBDM, system powinien poprosić o sterowniki do niego. Należy wybrać instalację z określonej lokalizacji i wskazać katalog **osbdm\_tbdml\osbdm\_driver** jako ścieżkę do sterowników. Urządzenie zostanie zainstalowane i będzie gotowe do pracy w środowisku CodeWarrior.

### 3.3 Interfejs Turbo BDM Light

Po podłączeniu interfejsu do komputera po jego wcześniejszym zaprogramowaniu jako TBDML, system powinien poprosić o sterowniki do niego. Należy wybrać instalację z określonej lokalizacji i wskazać katalog **osbdm\_tbdml\tbdml\_driver** jako ścieżkę do sterowników. Urządzenie zostanie zainstalowane i będzie gotowe do pracy w środowisku CodeWarrior.

### 3.4 Zworka zasilania

Na rysunku 4 przedstawiono opis zworki odpowiadającej za wybór źródła zasilania (Z5) dla translatorów napięć umieszczonych na liniach interfejsu BDM. Możliwe są do wyboru następujące opcje jej włożenia.

- 1 Zasilanie układu docelowego z interfejsu napięciem 5V (100mA)
- 2 Zasilanie interfejsu z układu docelowego napięciem z zakresu 1.2V-5V.
- 3 Zasilanie układu docelowego z interfejsu napięciem 3.3V (100mA)

Należy pamiętać, że jeśli programowany (debugowany) układ jest zasilany własnym zasilaniem nie należy zasilać go z programatora nawet, jeśli napięcie zasilania układu to jest równe 5 lub 3.3V. Należy w takim przypadku wybrać zasilanie zewnętrzne programatora (położenie **2** zworki Z5).

### 3.5 Zmiana firmware'u

W celu zmiany firmware'u urządzenia, chcąc zamienić typ interfejsu z tbdml na osbdm lub odwrotnie lub w celu wgrania nowszej wersji oprogramowania należy wypiąć zworkę Z3 i włożyć zworkę Z4 oraz ( jeśli są używane ) zworki Z1 i Z2. Urządzenie po ponownym podłączeniu do magistrali USB wejdzie w tryb programowania ICP i pozwoli wgrać nowy firmware jak fabrycznie nowy procesor.

Tablica 1: Spis elementów interfejsu TBDML i OsBDM

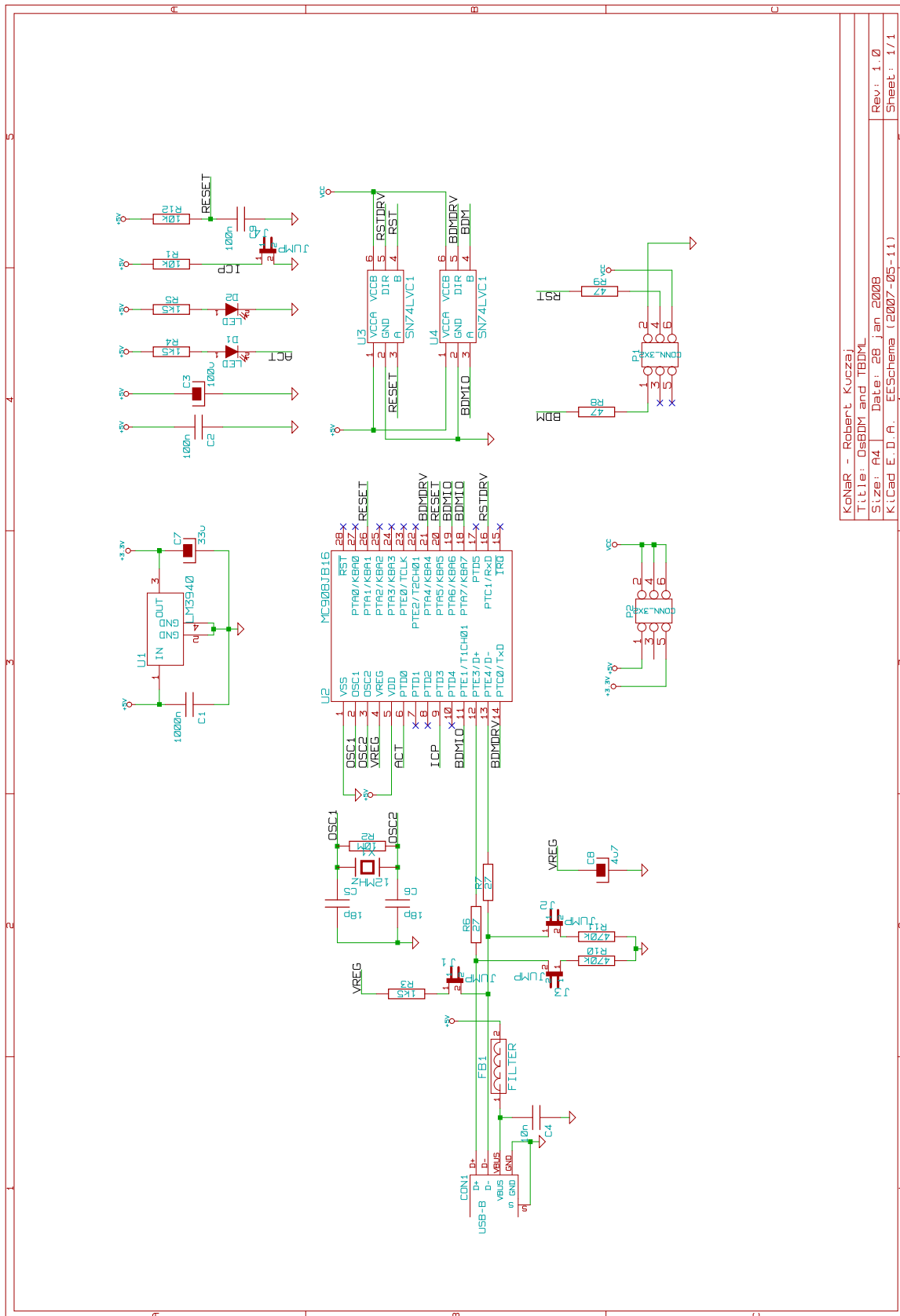
Symbol	Wartość	Obudowa	Producent	Dostawca	Kod u dostawcy
C1	1000nF	0805	-	Maritex	CMC0805-1000N-MY16
C2,C9	100nF	0805	-	Maritex	CMC0805-100N-MY50
C3	100uF	SMB	-	Maritex	ET100/
C4	10nF	0805	-	Maritex	CMC0805-10N-KX50-POE
C5,C6	18pF	0805	-	Maritex	CMC0805-18P
C7	33uF	SMB	-	Maritex	ET33/10B
C8	4,7uF	SMB	-	Maritex	ET4U7/25B
CON1 <sup>1</sup>	USB	THT	-	Maritex	USBS1BRB-LF
D1	zielona	φ3	-	Maritex	LED3G
D2	czerwona	φ3	-	Maritex	LED3R
FB1	Koralik ferrytowy	SMB	-	Maritex	EMISMB302520
J1,J2,J3,J4	goldpin 2x4	100"	-	Maritex	PLD20S-2
P1	Złącze IDC6 <sup>2</sup>	100"	-	Maritex	PCT10-LF
P2	goldpin 2x3	100"	-	Maritex	PLD20S-2
R1,R12	10kΩ	0805	-	Maritex	RMC0805-10K-Y
R2	10MΩ	0805	-	Maritex	RMC0805-10M-Y
R3,R4,R5	1,5kΩ	0805	-	Maritex	RMC0805-1K5-Y
R6,R7	27Ω	0805	-	Maritex	RMC0805-27R-Y
R8,R9	47Ω	0805	-	Maritex	RMC0805-47R-Y
R10,R11	470kΩ	0805	-	Maritex	RMC0805-470K-Y
U1	LM3940	SOT223	-	TME	LM3940IMP-3.3
U2	HC908JB16DWE	SOIC28W	Freescall	Microdis	MC908JB16DWE
U3,U4	SN74LVC1T45 <sup>3</sup>	SOT23-6	TI	Microdis	SN74LVC1T45DBVT
X1	12MHz	HC49S	-	Maritex	QS12M000
-	Taśma kablowa 10	50"	-	Maritex	RC10G-AWG28
-	Złącze IDC6	100"	-	Maritex	IDC06
-	3 zworki	100"	-	Maritex	MJ06BK4-LF
-	obudowa		-	TME	ABS-15N
-	kabel USB-A		-	TME	CAB-USB2AA/2
-	przepust kablowy	φ4	-	TME	M15059

## 4 Schemat układu i projekt PCB

Na rysunku 10 przedstawiony został schemat elektryczny opisanego interfejsu a na rysunku 11 widok projektu płytki drukowanej, na której został zmontowany. W tabeli 1 został przedstawiony szczegółowy spis elementów i ich pochodzenie.

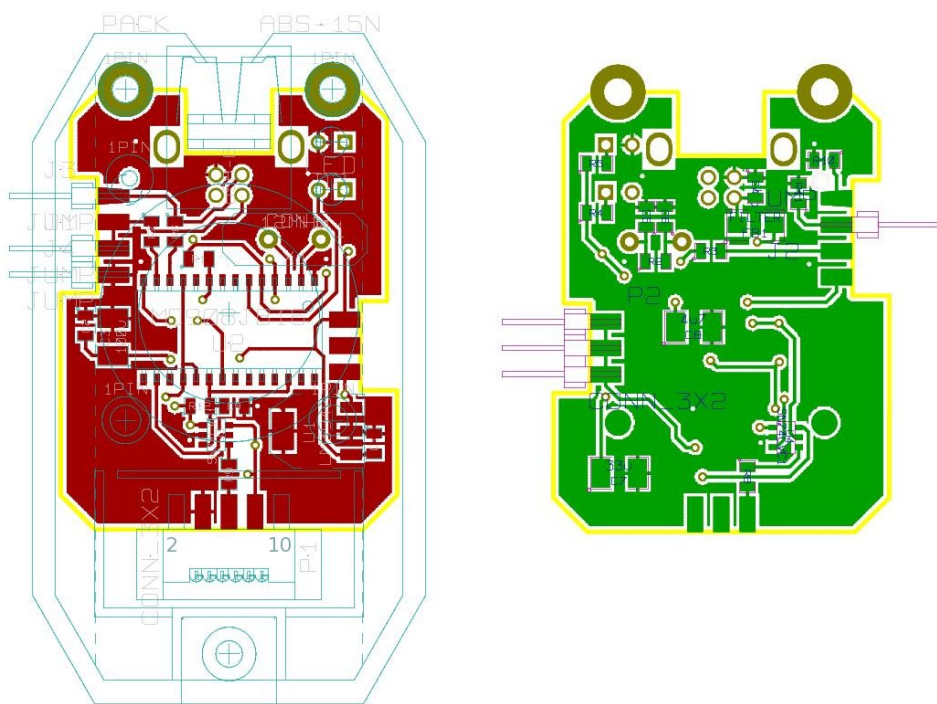
## Literatura

- [1] *Open Source BDM*, HCS08 BDM v1.0a, <http://www.systech-gmbh.ch/indexHCS08System.html>, 2006.
- [2] *Turbo BDM Light interface*, TBDML Manual r1.5, Daniel Malik, 2005.
- [3] *OSBDM forums*, <http://forums.freescale.com/freescale/board?board.id=OSBDM>, 2007.
- [4] *TBDML forums*, <http://forums.freescale.com/freescale/board?board.id=TBDML>, 2007.
- [5] *Freescall*, <http://freescale.com>.
- [6] *ICP Programing*, Application Note AN2399/D r0.3, <http://freescale.com>, 2003.



KoNaR - Robert Kuczaj  
 Title: OsBDM and TBDML  
 Size: A4 Date: 28 Jan 2008 Rev: 1.0  
 KiCad E.D.A. EESchema (2007-05-11) Sheet: 1/1

Rysunek 10: Schemat interfejsów TBDML i OSBDM



Rysunek 11: Widok projektu płytki interfejsów OSBDM i TBDML