

Na prawach rękopisu

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ
Raport serii SPR nr 19/2004

**Moduł z mikrokontrolerem
MC9S12C32**

Marek Wnuk

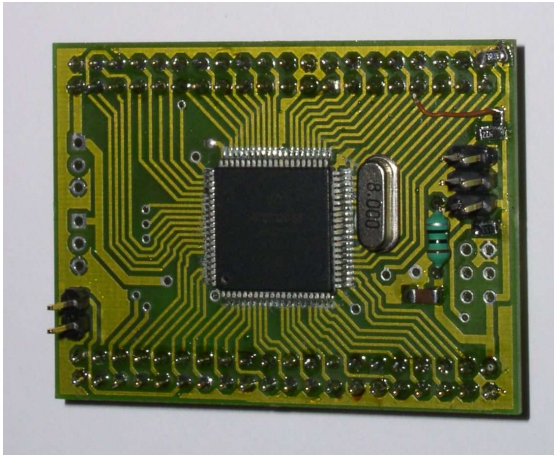
Słowa kluczowe: mikrokontroler, płytką drukowaną, system uruchomieniowy, oprogramowanie skróśne.

Wrocław 2004

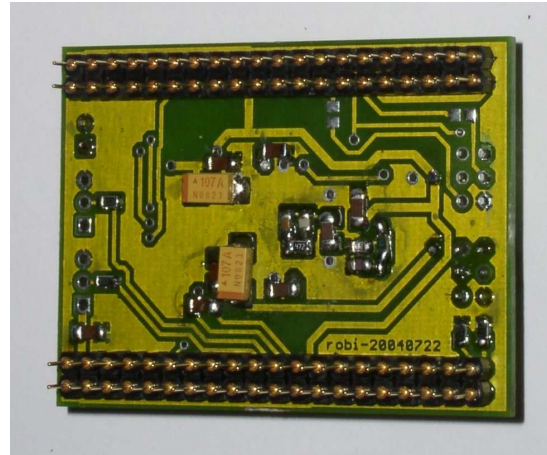
Spis treści

1	Wstęp	2
2	Własności mikrokontrolera MC9S12C32	2
3	Schemat i montaż	5
4	Złącza i sygnały	7
5	Uruchamianie sprzętu	9
6	Dokumentacje i oprogramowanie pomocnicze	10
	Bibliografia	10

widok z góry



widok z dołu



Rysunek 1: Wygląd zmontowanej płytki

1 Wstęp

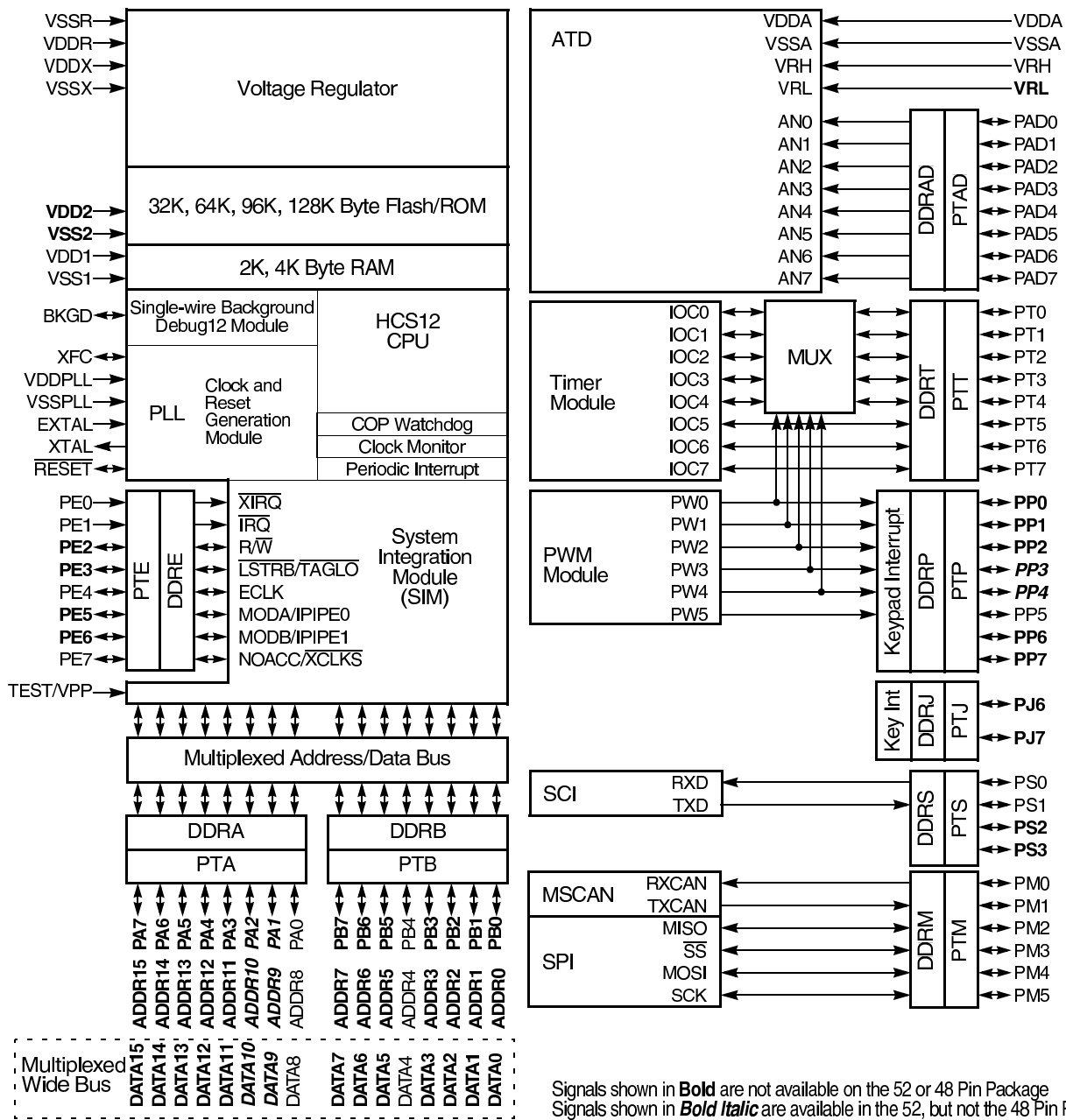
Opisywana płytki zawiera 16-bitowy mikrokontroler MC9S12C32 firmy Motorola oraz minimalny zestaw elementów towarzyszących (rezonator kwarcowy, zworki do wyboru trybu pracy, złącza BDM, rezystory wymuszające pożądany stan logiczny wybranych sygnałów i kondensatory blokujące zasilanie). Wszystkie sygnały mikrokontrolera są wyprowadzone na złącza w standardowym rozstawie (0.1 ″), co umożliwia dołączenie zewnętrznych rozszerzeń również przy użyciu uniwersalnej płytki drukowanej. Ma to szczególne znaczenie przy projektach badawczych i dydaktycznych, do których płytki jest przeznaczona. Widok modułu przedstawiono na rys. 1.

2 Własności mikrokontrolera MC9S12C32

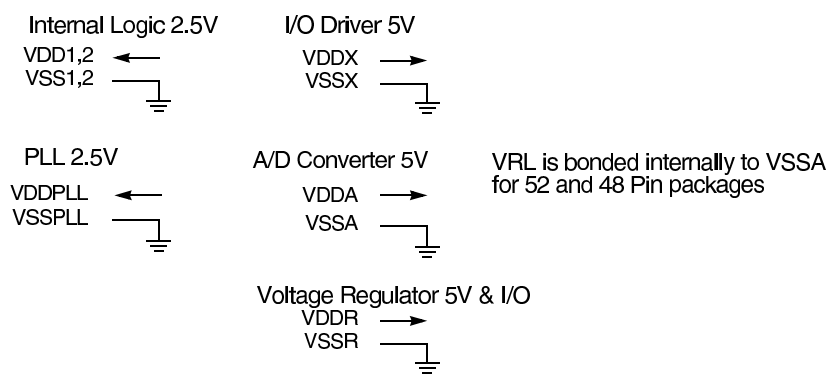
Rodzina HC12 mikrokontrolerów firmy Motorola jest następcą popularnej serii HC11, od wielu lat stosowanej w bardzo wielu konstrukcjach sterowników i układów sensorycznych, w szczególności w robotyce. Mikrokontrolery HC12 zostały wyposażone w bardzo wydajną, 16-bitową jednostkę centralną (CPU12) o wyższej częstotliwości taktowania, charakteryzują się małym poborem mocy oraz rozszerzonym w stosunku do HC11 zestawem układów wejścia/wyjścia (porty równoległe i szeregowo, timery i przetwornik analogowo–cyfrowy). CPU12 zachowuje zgodność z HC11 na poziomie asemblera, lecz ma poprawioną implementację wielu instrukcji. Wprowadzono też wiele dodatkowych instrukcji (np. interpolacji liniowej, *Fuzzy Logic*) oraz rozbudowane tryby adresowania indeksowego ułatwiają implementację systemów operacyjnych i języków wyższego rzędu (znane z M6809). Ważną zaletą CPU12 jest wbudowany emulator z interfejsem BDM pozwalający uruchamiać, testować i modyfikować oprogramowanie bezpośrednio na systemie docelowym.

Mikrokontroler MC9S12C32 jest istotnie ulepszony w stosunku do MC68HC912B32, który był wykorzystywany wcześniej [6]. Przy projektowaniu opisywanego poniżej modułu zadbano o zgodność rozmieszczenia wyprowadzeń mikrokontrolera na złącza zewnętrzne z modułem poprzednim, by ułatwić wymianę modułów w istniejących konstrukcjach.

Strukturę i zasoby mikrokontrolera MC9S12C32 [2] przedstawiono na rys. 2. Podstawowe własności zastosowanej wersji mikrokontrolera to:



Signals shown in **Bold** are not available on the 52 or 48 Pin Package
 Signals shown in **Bold Italic** are available in the 52, but not the 48 Pin Package



Rysunek 2: Struktura i zasoby MC9S12C32

- 16-bitowa jednostka centralna CPU12:
 - zgodność z listą rozkazów M68HC11,
 - analogiczny model programowy jak M68HC11,
 - 20-bitowa jednostka arytmetyczno-logiczna (ALU),
 - kolejka instrukcji,
 - rozbudowane tryby adresowania indeksowego,
 - wbudowane instrukcje *Fuzzy Logic*,
- zasoby pamięciowe:
 - 32 kilobajty wbudowanej pamięci FLASH (stronicowanie - 512bajtów),
 - 2048 bajtów wbudowanej pamięci RAM,
- 8-kanałowy, 10-bitowy przetwornik analogowo/cyfrowy z możliwością zewnętrznego wyzwala-
nia pomiaru,
- moduł transmisji CAN 2.0 A, B (5 buforów odbiorczych i 3 nadawcze, 4 x 16 lub 8 x 8 bitów, 4
oddzielne przerwania),
- 16-bitowy system timerów:
 - 16-bitowy licznik z 7-bitowym preskalerem,
 - 8 kanałów (każdy może być IC - *Input Capture* lub OC - *Output Compare*),
 - prosty tryb PWM,
 - możliwość resetowania licznika modulo,
- 16-bitowy akumulator impulsów:
 - zliczanie zdarzeń,
 - bramkowane zliczanie czasu,
- modulator PWM:
 - 4 kanały 8-bitowe lub 2 kanały 16-bitowe,
 - oddzielne sterowanie częstotliwością i wypełnieniem w kanałach,
 - tryb symetryczny (*Center-Aligned*) i asymetryczny (*Left-Aligned*),
 - wejście wyłączania awaryjnego,
- generacja zegara i reset:
 - układ przerwań cyklicznych o programowalnym okresie,
 - COP watchdog (*Computer Operates Properly*),
 - monitorowanie zegara procesora,
 - programowalny zegar procesora (PLL),
 - szybkość procesora do 50MHz (magistrala - 25MHz),

- interfejsy szeregowo:
 - asynchroniczny interfejs szeregowy NRZ (SCI),
 - synchroniczny interfejs szeregowy urządzeń zewnętrznych (SPI),
- 61 uniwersalnych wejść-wyjść binarnych:
 - 51 dwukierunkowych,
 - 10 wyłącznie wejściowych,
- wbudowany regulator napięcia 2.5V
 - zasilanie zewnętrzne 3.3V - 5.5V,
 - reset przy zbyt niskim napięciu (LVR),
 - przerwanie przy zbyt niskim napięciu (LVI),
- interfejs uruchomieniowy (wbudowany emulator) BDM (*Background Debug Mode*) z pułapkami sprzętowymi,
- obudowa QFP80,

3 Schemat i montaż

Schemat modułu przedstawiono na rys. 3, a sposób montażu na płytce drukowanej na rys. 4¹. Na płytce zamontowano: mikrokontroler MC9S12C32 w obudowie QFP80 (U1), rezonator kwarcowy 8MHz (Q1) wraz z elementami towarzyszącymi (C1, C2 - 22pF, R1 - 10MΩ), kondensatory blokujące zasilanie (C3 .. C8, C14, C15 - 100nF, C10, C11 - 100μF), rezystor wymuszający stan wysoki sygnału restartu (RESET) i stan niski sygnałów (MODA, MODB) wyboru trybu CPU (R6, R13, R15 - 15kΩ). Elementy: C12 (330pF), C13 (3.3nF) i R8 (3.9kΩ) tworzą filtr w układzie generacji zegara (PLL). Zasilanie przetwornika C/A jest dodatkowo filtrowane przez dławik L1 (10 μH).

Uwaga: PE7/NOACC/XCLKS (SVL.1) należy połączyć z masą (Vss) przez rezystor 3k9, by zapewnić uruchomienie generatora zegara po restarcie w odpowiednim trybie. To połączenie nie jest zrealizowane na płytce drukowanej.

W celu zachowania zgodności z modułem uruchomieniowym M68EVB912B32 firmy Motorola [1] i umożliwienia wykorzystania monitora/debuggera D-Bug12 do uruchamiania modułu wprowadzono dwa sześciostykowe złącza BDM-IN i BDM-OUT. W drugim z nich sygnały PT6 i PT7, za pośrednictwem odsprzęgających rezystorów R4 i R5 (47Ω), są użyte do emulowania sygnałów BKGD i RESET dla zewnętrznego układu docelowego (przy wykorzystaniu modułu jako interfejsu BDM). Wybór trybu pracy D-Bug12 zapewniają zworki J1 i J2 z rezystorami R2 i R3 (15kΩ) podłączonymi do sygnałów PAD0 i PAD1 mikrokontrolera. W przypadku nie korzystania z D-bug12 można nie montować: J1, J2, J4, R2 – R5.

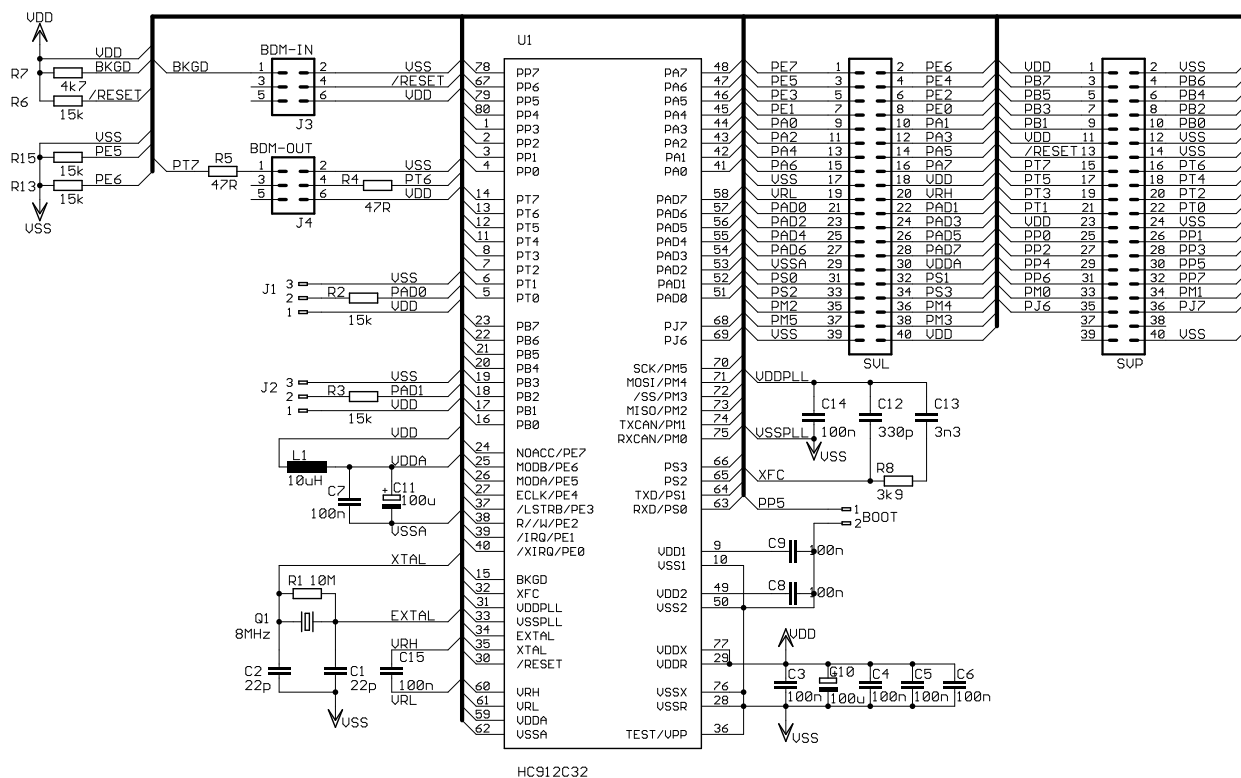
Aby umożliwić wykorzystanie monitora/debuggera S12SerMon [5] wprowadzono zworkę BOOT pozwalającą wymusić stan niski na linii portu PP5.

Mikrokontroler, rezonator kwarcowy, dławik, złącza BDM i zworki są zamontowane na wierzchniej stronie płytki. Pozostałe elementy są montowane na dolnej powierzchni modułu.

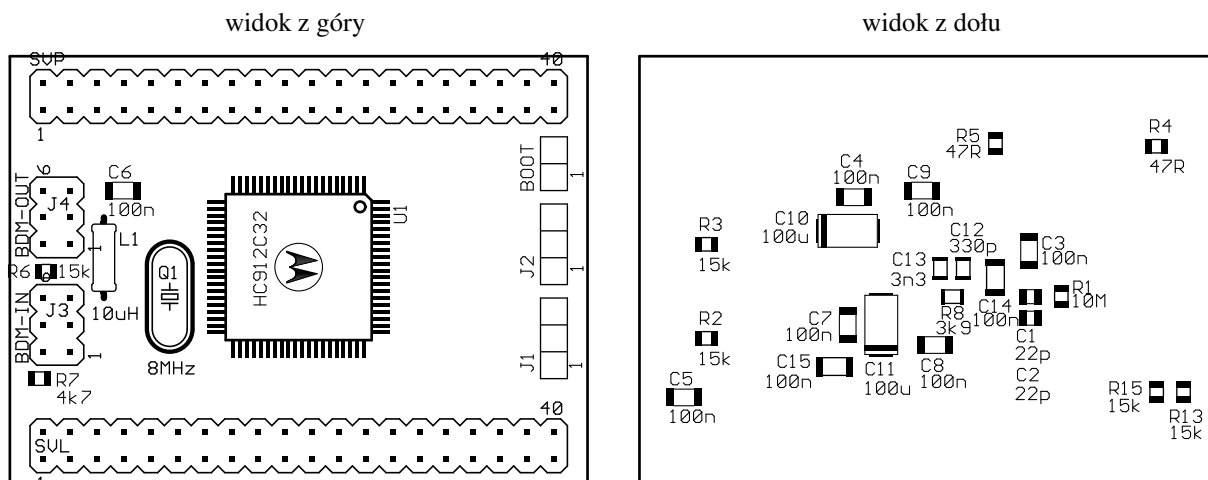
¹Projekt płytki drukowanej wykonał Robert Szlowski (Laboratorium Robotyki ICT PWr.).

liczba	wartość	obudowa	oznaczenie na schemacie
rezystory			
2	47R	0805	R4, R5
1	3k9	0805	R8
1	4k7	0805	R7
5	15k	0805	R2, R3, R6, R13, R15
1	10M	0805	R1
kondensatory			
2	22p	0805	C1, C2
1	330p	0805	C12
1	3n3	0805	C13
9	100n	1206	C3, C4, C5, C6, C7, C8, C9, C14, C15
2	100u/6,3V	6032	C10, C11
filtry			
1	10uH	0207	L1
rezonatory			
1	8MHz	HC49/S	Q1
półprzewodniki			
1	MC9S12C32	QFP80	U1

Tablica 1: Zestawienie elementów modułu MC9S12C32



Rysunek 3: Schemat układu



Rysunek 4: Montaż elementów na płytce

4 Złącza i sygnały

Sygnały mikrokontrolera wyprowadzono na dwa dwurzędowe złącza (SVL, SVP) zawierające po 40 styków. Użycie listew wtykowych SLP zamontowanych od spodniej strony płytki pozwala włączyć moduł do odpowiednich gniazd zamontowanych na innej płytce. Możliwe jest również użycie dwurzędowych listew stykowych zamontowanych od wierzchniej strony płytki w celu dokonania połączeń zewnętrznych przewodem taśmowym.

złącze SVL				złącze SVP			
PE7/NOACC/XCLKS	1	2	PE6/MODB	VDD(+5V)	1	2	VSS(GND)
PE5/MODA	3	4	PE4/ECLK	PB7	3	4	PB6
PE3/LSTRB	5	6	PE2/RW	PB5	5	6	PB4
PE1/IRQ	7	8	PE0/XIRQ	PB3	7	8	PB2
PA0	9	10	PA1	PB1	9	10	PB0
PA2	11	12	PA3	VDD(+5V)	11	12	VSS(GND)
PA4	13	14	PA5	RESET	13	14	VSS(GND)
PA6	15	16	PA7	PT7/IOC7/PAI	15	16	PT6/IOC6
VSS(GND)	17	18	VDD(+5V)	PT5/IOC5	17	18	PT4/IOC4
VRL	19	20	VRH	PT3/IOC3	19	20	PT2/IOC2
PAD0/AN0	21	22	PAD1/AN1	PT1/IOC1	21	22	PT0/IOC0
PAD2/AN2	23	24	PAD3/AN3	VDD(+5V)	23	24	VSS(GND)
PAD4/AN4	25	26	PAD5/AN5	PP0/PW0	25	26	PP1/PW1
PAD6/AN6	27	28	PAD7/AN7	PP2/PW2	27	28	PP3/PW3
VSSAD	29	30	VDDAD	PP4/PW4	29	30	PP5/PW5
PS0/RxD	31	32	PS1/TxD	PP6	31	32	PP7
PS2	33	34	PS3	PM0/RxCAN	33	34	PM1/TxCAN
PM2/SDI/MISO	35	36	PM4/SDO/MOSI	PJ6	35	36	PJ7
PM5/SCK	37	38	PM3/CS/SS	n.c.	37	38	n.c.
VSS(GND)	39	40	VDD(+5V)	n.c.	39	40	VSS(GND)

Tablica 2: Złącza modułu MC9S12C32

Rozmieszczenie sygnałów na złączach SVL i SVP podano w tab. 2, a ich opis w tab. 3. Sygnały na złączach BDM opisano w tab. 4. W przypadku BDM-IN sygnały RESET i BKGD są sy-

sygnał	wyprowadzenie	typ	opis
PA0-PA7	SVL.9-SVL.16	IO	port A.[0-7]
PAD0/AN0-PAD7/AN7	SVL.21-SVL.28	I	wejście analogowe [0-7]
VRL	SVL.19	PWR	dolne napięcie odniesienia ADC
VRH	SVL.20	PWR	górne napięcie odniesienia ADC
PB0-PB7	SVP.10-SVP.3	IO	port B.[0-7]
PJ6-PJ7	SVP.35-SVP.36	IO	port J.[6-7]
PE0/XIRQ	SVL.8	I	przerwanie niemaskowalne
PE1/IRQ	SVL.7	I	przerwanie maskowalne
PE2/RW	SVL.6	IO	kierunek danych
PE3/LSTRB	SVL.5	IO	strob dolnego bajtu
PE4/ECLK	SVL.4	IO	zegar magistrali
PE5/MODA	SVL.3	IO	wybór trybu pracy CPU12
PE6/MODB	SVL.2	IO	wybór trybu pracy CPU12
PE7/NOACC/XCLKS	SVL.1	IO	wybór trybu pracy ocylatora
PM0/RxCAN	SVP.33	IO	dane odbierane CAN
PM1/TxCAN	SVP.34	IO	dane nadawane CAN
PM2/SDI/MISO	SVL.35	IO	dane SPI <i>Master In Slave Out</i>
PM3/CS/SS	SVL.38	IO	we/wy wyboru SPI <i>Slave Select</i>
PM4/SDO/MOSI	SVL.36	IO	dane SPI <i>Master Out Slave In</i>
PM5/SCK	SVL.37	IO	zegar SPI
PP0/PW0-PP5/PW5	SVP.25-SVP.28	IO	wyjście modulatora PWM [0-5]
PP6-PP7	SVP.29-SVP.32	IO	port P.[6-7]
PS0/RxD	SVL.31	IO	dane odbierane SCI
PS1/TxD	SVL.32	IO	dane nadawane SCI
PS2-PS3	SVL.33-SVL.34	IO	port PS.[2-3]
PT0/IOC0-PT6/IOC6	SVP.22-SVP.16	IO	we/wy <i>Input Capture / Output Compare</i> [0-6]
PT7/IOC7/PAI	SVP.15	IO	we/wy <i>Input Capture / Output Compare</i> 7 wejście <i>Pulse Accumulator</i>
RESET	SVR.13	IO	restart
VSS(GND)	SVL.17, SVL.39, SVP.2, SVP.12, SVP.14, SVP.24, SVP.40	PWR	masa
VDD(+5V)	SVL.18, SVL.40, SVP.1, SVP.11, SVP.23	PWR	zasilanie (+5V)

Tablica 3: Sygnały modułu MC9S12C32

BKGD	1	2	VSS(GND)
n.c.	3	4	RESET
n.c.	5	6	VDD(+5V)

Tablica 4: Złącza BDM modułu MC9S12C32

gnałami wejściowymi mikrokontrolera modułu. W przypadku BDM-OUT są one wytwarzane (emulowane) przez D-Bug12 za pośrednictwem sygnałów PT6 i PT7 mikrokontrolera.

5 Uruchamianie sprzętu

W celu zaprogramowania w pamięci FLASH zmontowanego modułu oprogramowania firmowego (D-Bug12) należy się posłużyć interfejsem BDM i odpowiednim oprogramowaniem na komputerze nadrzędnym. W najprostszym przypadku rolę interfejsu BDM może pełnić inny, wcześniej uruchomiony i oprogramowany, moduł (np. płytką M68EVB912B32 firmy Motorola) ustawiony w tryb 01 (*POD – Remote Debugging Through BDM-OUT* [1]). W takim przypadku należy połączyć sześciopiętowym kablem taśmowym złącze BDM-OUT płytki służącej jako interfejs BDM ze złączem BDM-IN uruchamianego modułu. Na komputerze nadrzędnym należy uruchomić emulator terminala szeregowego i wykorzystać port szeregowy komputera do komunikacji z D-Bug12 (9600, 8, N, 1). Używając komend debuggera [1] można skasować pamięć FLASH uruchamianego modułu (komenda FBULK) i zaprogramować ją nową zawartością podaną w pliku z S-rekordami (komenda FLOAD). Aktualną wersję D-Bug12 można znaleźć na stronie <www.mot.com>.

Dzięki zworkom J1 i J2 można wybrać jeden z czterech trybów pracy D-Bug12 w opisywanym module:

- normalny (*EVB Execution Mode*) – D-Bug12 z pamięci FLASH jest po restarcie uruchamiany interakcyjnie ze znakiem zachęty “>” na terminalu dołączonym do SCI,
- start z pamięci EEPROM (*Jump-EEPROM Mode*) – po restarcie jest uruchamiany program użytkownika zapisany w pamięci EEPROM,
- zdalne debuggowanie przez BDM-OUT (*POD Mode*) – D-Bug12 łączy się ze zdalnym modulem przez BDM i zgłasza znak zachęty “S>”,
- ładowanie programu (*BOOTLOAD Mode*) – program użytkownika w postaci S-rekordów jest ładowany do pamięci FLASH i/lub EEPROM.

zworki		tryb uruchamiania
J1	J2	D-Bug12 po restarcie
2-3	2-3	EVB Execution
1-2	2-3	Jump-EEPROM
2-3	1-2	POD
2-3	2-3	BOOTLOAD

Tablica 5: Zworki wyboru trybu startowego D-Bug12

W tab. 5 podano ustawienia zworek J1, J2 odpowiadające opisanym trybom. Dokładny opis tych trybów i zestawienie komend D-Bug12 można znaleźć w [1]).

Wygodnym środkiem do uruchamiania oprogramowania użytkowego na mikrokontrolerach z rodziny HCS12 jest 2k-bajtowy monitor S12SerMon. Obsługuje on 23 komendy pozwalające na programowanie pamięci FLASH/EEPROM i uruchamianie programów przez łącze szeregowe. Komendy obejmują odczyt i modyfikowanie zawartości pamięci (również FLASH i EEPROM), odczyt i zapis rejestrów procesora, uruchamianie, zatrzymywanie i krokowe wykonywanie programu. Obszar pamięci FLASH (2 kilobajty: \$F800-\$FFFF), w którym rezyduje monitor, jest zabezpieczony przed zapisem. Zmiana zawartości tego obszaru jest możliwa tylko przy użyciu interfejsu BDM. Obsługa przerwań

jest przekierowana do obszaru niechronionego, co pozwala użytkownikowi podłożyć własne procedury. Monitor nie używa pamięci RAM poza stosem. Dokładny opis monitora i jego interfejsu z komputerem nadrzędnym (PC) można znaleźć w [5] (dokument jest dostępny na <www.freescale.com>). Przykładem aplikacji systemu Win32 jwspółpracującej z opisanym monitorem jest uBug12 firmy Technological Arts (<www.technologicalarts.com>), dostarczany łącznie z zestawem uruchomieniowym M68DM912C32 [4].

UWAGA: w celu połączenia portu szeregowego SCI opisywanego modułu z komputerem nadrzędnym wyposażonym w port szeregowy RS232C należy użyć układu translacji napięć z TTL do RS232C (np. MAX232).

6 Dokumentacje i oprogramowanie pomocnicze

Dokumentacje mikrokontrolera MC9S12C32 oraz całej rodziny HCS12 są dostępne na serwerze firmy Freescale (d. Motorola Semiconductors – <www.freescale.com>). Na lokalnym serwerze Laboratorium Robotyki ICT PWr. (<rab.ict.pwr.wroc.pl>) są dostępne ich kopie zamieszczone za zgodą Motorola Polska, jak również wybrane narzędzia programistyczne i przykłady zastosowań. Oprogramowanie dla CPU12 przygotowuje się w języku asemblera [3]. Dostępne są proste skrótnie asemblery dla środowiska DOS/Windows, zarówno darmowe (<www.mot.com>), jak i komercyjne. Przykładem zintegrowanego środowiska dla Windows, obejmującego edytor, asembler i program komunikacyjny jest IASM12 firmy P&E (<www.pemicro.com>)

Bardziej rozbudowane środowiska, obejmujące również kompilatory C/C++ to pakiety darmowe (GNU) i komercyjne (np. CodeWarrior firmy Metrowerks – <www.metrowerks.com>). Odwołania do stron WWW zawierających informacje o takich pakietach można znaleźć na stronie laboratorium: <rab.ict.pwr.wroc.pl>.

Literatura

- [1] *M68EVB912B32 Evaluation Board User's Manual*, 68EVB912B32UM/D, Motorola Inc. 1997.
- [2] *MC9S12C Family Device User Guide*, 9S12C128DGV1/D Rev. 01.10, Motorola Inc. 2004.
- [3] *S12CPUV2 Reference Manual*, S12CPUV2/AD Rev. 2.0, Motorola Inc. 2003.
- [4] *Using Your M68DKIT912C32 Microcontroller Kit*, M68DKIT912C32UM, Technological Arts, <www.technologicalarts.com>, 2004.
- [5] Williams J., *Serial Monitor Program for HCS12 MCUs*, AN2548/D, Application Note, Motorola Inc. 2003.
- [6] Wnuk M., *Moduł z mikrokontrolerem MC68HC912B32*, Raport ICT PWr. serii SPR 6/04, Wrocław, 2004.

dr inż. Marek Wnuk
Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej
ul. Janiszewskiego 11/17
50-372 Wrocław

Niniejszy raport otrzymują:

- 1. OINT - 1 egz.
- 2. Zleceniodawca - 1 egz.
- 2. Autor - 3 egz.

Razem : 5 egz.

Raport wpłynął do redakcji I-6
w grudniu 2004 roku.