AVR JTAG-ICE Interfejs debugera dla mikrokontrolerów AVR



Bardzo przydatnym, a w niektórych sytuacjach wrecz niezbednym narzędziem do uruchamiania systemów mikroprocesorowych jest debuger. Dla mikrokontrolerów AVR dobrym i tanim rozwiązaniem jest AVR Dragon. Niestety, jeśli ulegnie on uszkodzeniu, to naprawa jest najczęściej nieopłacalna. Na stronie firmy Atmel umieszczono opis programatora/ debugera zgodnego z JTAG-ICE, który można wykonać samodzielnie, ale jego wadą jest skomplikowana budowa. W artvkule zaprezentowano opis programatora/debugera zgodnego *i* JTAG-ICE o nieskomplikowanej konstrukcji. Ponadto, wyposażono go w bufory I/O, dzięki czemu szanse uszkodzenia programatora są dość małe, a jeśli uda nam się coś zepsuć, to koszt naprawy jest bardzo niski. Rekomendacje: JTAG przyda się każdemu zajmującemu się programowaniem lub uruchamianiem układów wyposażonych w mikrokontrolery AVR.

Na projekt programatora natknąłem się na stronie http://www.m2uu.com/elektronika-:avrjtag. Jednak aby nieco uodpornić JTAG na różne sytuacje, które zdarzają się podczas uruchamiania urządzeń z mikrokontrolerami, zdecydowałem się na zmodyfikowanie oryginalnej konstrukcji. W porównaniu z oryginałem wprowadziłem następujące zmiany:

- dodałem bufor 74HC244 zabezpieczony rezystorami szeregowymi,
- konwerter USB-RS232 zasiliłem z magistrali USB.
- pozostałe układy programatora zasiliłem z uruchamianego systemu, dzięki czemu



programator może pracować zasilany napięciem innym niż 5 V.

Budowa i zasada działania

Schemat ideowy JTAG'a pokazano na rysunku 1. Do połączenia z komputerem służy interfejs USB. Komunikacja odbywa się poprzez UART za pośrednictwem popularnego układu konwertera, układu U1 typu FT232RL. Jest on zasilany z portu USB. Pozostałe obwody programatora są zasilane z uruchamianego systemu, dlatego wyprowadzenie VccIO układu U1 dołączono do wyprowadzenia 4 złącza JP3, a nie do portu USB.

Mikrokontroler U2 przyjmuje polecenia od komputera PC i steruje interfejsem JTAG, który jest emulowany przez port SPI. Wszystkie linie JTAG są buforowane układem U3 typu 74HC244. Sam bufor dodatkowo zabezpieczono rezystorami szeregowymi. Dla wejść są to rezystory o rezystancji 1 kΩ, natomiast dla wyjść 100 Ω. Dzięki temu podanie napięcia w zakresie -20...+25 V na wejście nie powinno spowodować uszkodzenia bufora (prąd wejścia nie przekroczy 20 mA). Wyjście jest zabezpieczone przed napięciami w zakresie -2 V...Vcc+2 V. A jeśli mimo wszystko bufor uszkodzi się, łatwo go wymienić, ponieważ jest zamontowany w podstawce.

W stosunku do rozwiązania dostępnego w Internecie zmieniłem wartość rezystora R6 w dzielniku napięcia. Dodałem także źródło napięcia odniesienia 2,5 V. Spowodowane było to tym, że mikrokontroler jest zasilany

AVT-5322 w ofercie AVT: AVT-5322A – płytka drukowana AVT-5322B – płytka drukowana + elementy

Podstawowe informacje: • Lista obsługiwanych układów: ATmega128, ATmega128L, AT90CAN128, ATmega64, ATmega64L, ATmega32, ATmega32L, Armegaðat, Armegað23, Armegað24, Armegað26, Armega162, Armega162, Armega162, Armega162V, Armega165, Armega162V, Armega169, Armega169L, Armega169V

- Zasilanie JTAG'a z uruchamianego urządzenia.
 Napięcie pracy 2,7...5,25 V.
- Współpraca z AVR Studio.

Dodatkowe materiały na CD/FTP: tp://ep.com.pl, user: 17692, pass: 4yv87ftn
wzory płytek PCB
karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie

elementów kolorem czerwonym

Projekty pokrewne na CD/FTP:

	Jenty pometric na es/ini
(wymie	nione artykuły są w całości dostępne na CD)
AVT-5279	Programator PIC (EP 2/2011)
AVT-5172	Uniwersalny programator
	mikrokontrolerów AVR (EP 2/2009)
AVT-5153	Uniwersalny programator JTAG/ISP
	(EP 10/2008)
AVT-5125	Programator USB AVR (STK500)
	(EP_2/2008)
AVT-1462	Uniwersalny adapter dla
	programatorów AVR-ISP (EP 2/2008)
AVT-2855	Ulepszony programator STK200
	(EdW 2/2008)
AVT-988	Programator AVRISP z interfejsem
	USB (STK500) (EP 7/2007)
AVT-1452	Adapter dla programatorów AVR ISP
	(EP 7/2007)
AVT-947	Programator JTAG dla
	mikrokontrolerów STR9 (EP 9/2006)
AVT-937	Programator ISP/ICP dla
	mikrokontrolerów ST7 (EP 7/2006)
AVT-921	Flash z ISP – JTAG (EP 3/2006)
AVT-451	Programator z interfejsem USB dla
	Bascom AVR (EP 11/2005)
AVT-1409	Programator JTAG dla układów
	MSP430 (EP 3/2005)
AVT-540	Miniprogramator AT89Cx051
	(EP 11/2004)

z uruchamianego systemu, a napięcie odniesienia JTAG-ICE pochodzi z wejścia Aref. Gdyby mikrokontroler był zasilany napieciem 5 V, to pomiar napiecia występujacego na złączu JTAG byłby prawidłowy. Niestety, konieczne byłby zasilanie JTAG'a z jakiegoś stabilizatora, ponieważ złącze USB komputera PC nie gwarantuje napięcia zasilającego 5 V. Może ono wynosić np. tylko 4,4 V. Dzięki podzieleniu napięcia ze złącza JTAG przez 2 w stosunku do orvginału (rezystor R6 o rezystancji 75 k Ω , a nie 150 k Ω) oraz zastosowa-

Gdy brak U4, to R6=150k

U2 AtMega16A-AU

32

98

38 38 30

14

45 43

NTRST

×SMT≯

ŝ

Vref Vref

ö⊲

UND

178 75

TARN tt

с ц

U4 LM385-2.5

- N 0

8

PA5/ADC5 PA6/ADC6

PA4/ADC4

ž⊩

-23

PA 3/A D C3

PA 2/A DC2

PA 0/A D C0

PB0/XCK/T0

STNI/0NIA\289 17\189

PB3/AIN1/OC0

<u>NSO</u>

bufor zawsze aktywny

J3: brak zwory – normalna praca, bufor zawsze aktywn; 2–3 – normala praca, bufor aktywny w czasie transr 1–2 – wgywanie programu BootLoadera przez ISP

ξ ģ

NOR BOD

R20

R19 OOR

73

£∏¥

£₹

20 V

GND

SS/78d

niu napięcia Aref o wartości 2,5 V zamiast 5 V, pomiary napięcia wykonywane przez AvrStudio sa prawidłowe. Możliwe jest oczywiście zrezygnowanie z U4. Wtedy rezystor R6 musi mieć rezystancje 150 k Ω .

Uwaga! Jeśli nie korzystamy z U4 (R6 =150 kΩ), pomiar może być obarczony dużym błędem. Wynika to z faktu, że napięcie w złączu USB może się zawierać w granicach 4,4...5 V. W skrajnym przypadku (4,4 V) bład wynosi +12%. W takiej sytuacji faktyczne napięcie 5 V będzie przedstawio-

ne jako 5,6 V, natomiast 3,3 V jako 3,69. Do błędu tego należy doliczyć jeszcze tolerancję rezystorów R6 i R7. W prototypie bez U4, 5 V było obrazowane w AvrStudio jako 6,2 V!

Montaż i uwagi odnośnie do komponentów

Schemat montażowy JTAG'a pokazano na rysunku 2. Montaż jest typowy i nie wymaga szczegółowego opisu. Na początku warto wlutować układy SMD znajdujące się od strony ścieżek. Wlutowanie U2 nie sprawi wiekszego kłopotu, natomiast U1 jest już bardziej kłopotliwy w montażu ze względu na mały raster wyprowadzeń. Osobiście z U1 poradziłem sobie w ten sposób, że po pozycjonowaniu układu i przylutowaniu skrajnych nóżek, zalałem stopem lutowniczym wszystkie nóżki, a następnie nadmiar cyny zebrałem za pomocą plecionki "WIK". Po wlutowaniu elementów SMD montujemy zwory. Pod U3



poziomu AvrStudic

-

R9 ¥₩ R1 133 Å 14 - N O NTRST NSRT MISO MOSI IMS WORK/Prog SP SCK Rysunek 1. Schemat ideowy JTAG'a AVR

202

- R

δ

GND

JSBB-BV

-∎₽

ç

H Å





Rysunek 2. Schemat montażowy JTAG'a AVR

należy zastosować podstawkę. Płytka jest przystosowana do umieszczenia w obudowie KM-35.

Jako układ U3 można zastosować 74HC244 lub 74AC244. Nie mogą to być układy typu HCT lub ACT, ponieważ nie będą one pracować przy napięciach różniących się od 5 V o więcej niż 10%. Nie może to też być układ LS, nawet gdy programator miałby pracować przy napięciach 5 V, a to ze względu na prąd wejścia (konieczna byłaby korekta wartości rezystorów zabezpieczających).

Jako U2 można użyć AtMega16A-AU lub AtMega16L-8AU. Nie może to być At-Mega16-AU, ponieważ napięcia pracy zo-

		P = P ? 0	
Besic Details Device Type FT JSB VID /PID FT Vencer ID 5403 BM / C Device Spe USB Vension Num F Disable USB S F Aut Down10 P Disable USB S Product and Menuf	2228 V Et Detaut Product D 0001 offic Options erer 1000 22 V erei Nuchter the huldes huldes r Description Strings	USD Renck Option	PT223R PT223R PT422 Invert R5222 Signals Invert R522 Signals Invert R52 Invert R52 Invert R53 Invert R53 Invert R54 Invert R54 Invert R55 Invert R54 Invert R54 Invert R54 Invert R54
FTD	FT232R USB UAR	T	SLEEV C4
Programming Optio	ns ank Devices		F High Current IKO's Load D2XX driver

Rysunek 3. Ustawienie funkcji wyprowadzenia C0



Rysunek 4. Ustawienie fuse bitów mikrokontrolera za pomocą PonyProg

staną ograniczone do zakresu 4,5...5,25 V (górną granicę napięcia wprowadza układ U1).

Uruchomienie

W pierwszej kolejności należy podłączyć nasze urządzenie do portu USB komputera (wystarczy zasilanie z USB). Sposobu instalacji sterowników nie będę opisywał, ponieważ ten temat był już poruszany wielokrotnie. Ze względu na zastosowanie układu FT232RL można przy użyciu programu "MProg.exe", ustawić funkcje wyprowadzenia C0 na "TX & RX LED# (**rysunek 3**).

Następnym krokiem będzie zaprogramowanie U2, o ile nie korzystamy z już zaprogramowanego mikrokontrolera. Na

Fuse	Value
M103C	
WDTON	
OCDEN	•
JTAGEN	v
SPIEN	V
EESAVE	Π
BOOTSZ	Boot Flash size=4096 words start address=\$F000
BOOTRST	-
CKOPT	П
BODLEVEL	Bassen-out detection level at VCC+2.7 V
DODEN	
	>
EXTENDED	0xFF
HIGH	0x18
LOW/	DIFF
Auto read	
Smart warrings	
Verile alter reco	Program Verily Read
- I ronly and prog	
aving programming	mode. 017
aving programming R Dragon in JT	mode. 013 AG mode with ATmega16
aving programming R Dragon in JT ain Program	mode. 007 AC mode with ATmrgs16
aving programming R Dragon in JT ain Program	node. 017 AG mode with ATmrga16 Fase: LockSte Advanced Hv/ Settings Hv/ Into Auto Vale
R Dragon in JT ain Program	AG mode with ATmega16
R Dragon in JT ain Program	AG mode with ATmega15 Add mode with ATmega15 Faste Lookie Advanced HW Setting MV Into Adva Vala
R Dragen in JT ain Program Fuse OCDEN JTAGEN SPIEN	AG mode with ATrings 16.
R Dragon in JT ain Program Fute OCDEN JTAGEN SPIEN EESAVE	AG mode with ATmega16
R Dragon in JT ain Program I Fuse OCDEN JTAGEN SPIEN EESAVE BOOTS2	AG mode with ATmrga16 AG mode with ATmrga16 Value Value Bo Plank iden1024 words nat address=\$100
R Dragen in JT ain Piogram Fute OCDEN JTADEN SPIEN EESAVE BOOTRST	AG mode with ATmenga16 AG mode with ATmenga16 Weile Value Bon Rhan bice=1024 words that address=4100
R Dragen in JT ain Program Fuse OCDEN JTAGEN SPIEN EESAVE BOOTS2 BOOTRST CXOPT	AC mode with ATmrga16 AC mode with ATmrga16 Advanced Hw/Sellings Hw/Info Auto Value Boo Flash ide=1004 worth stat address=\$1000 C
R Dragon in JT ain Program T Fuse OCDEN JTAGEN SPIEN EESAVE BOOTS2 BOOTS2 BOOTS5 CXOPT BODLEVEL	AG mode with ATmrga16 AG mode with ATmrga16 AG mode with ATmrga16 Wate LooBin Advanced Hw Settings Hw Into Auto Wate Bit Control (Control (Contro) (Control (Control (C
R Dragon in JT ain Program Fute OCDEN JTAGEN SPIEN EESAVE BOOTS2 BOOTS2 BOOTS2 BOOTS2 CXOPT BODLEVEL BODEN	AC mode with ATmega16 AC mode with ATmega16 Action and the strength of the
R Dragon in JT ain Program Fuse OCDEN JTADEN SREN EESAVE BOOTRST CXOPT BODLEVEL BODEN SUT_CXSEL	AG mode with A Tringg110 Ag mode with A Tri
R Dragen in JT ain Program Fue OCDEN JTACEN SPIEN EESAVE BOOTS2 BOOTS2 BOOTS2 BOOTS2 BOOTS2 BOOTS4 BOOLEVEL BODEN SUT_CKSEL HIGH	AG mode with A Tringg10 AG mode with A Tringg10 AG mode with A Tring
Crappo II JT Am Program Fue OCCEN JTAGEN SPEN EESANE EESANE EESANE BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ BOOTRSZ HIGH LOW	AG mode with XTranga15 Add mode with XTranga16 Add mode with XTranga16
Craigen in JT Anagen in JT Anagen Program Program Program Prove OCOEN JTAGEN SPEN EESAVE BOOTS2 BOOTS2 BOOTS2 BOOTS2 BOOTS4 CXOPT BOOLVEL BOOEN SUT_OKSEL HIGH LOW	AG mode with XTranga1.6 AG mode with XTranga1.6 Value Value Exc Plank nice-1024 words start addees-4100 Exc Plank nice-1024 words start addees-4100 Exc Plank nice-1024 words start addees-4100 Common ad detection at VCC-27 V Exc Plank Resonance High Freq: Startup line: 195.01 - 64 re Dole Dole
R Dragon in JT ain Pogram Fue OCDEN JTAGEN SPIEN EESAVE BOOTRST ENOLYEL BOOEN SUT_CKSEL	AG mode with A Transp16 AG mode with A Transp16 Weile Vale Biol Rath los-1024 words that address-\$1000 Biol Rath los-1024 words that address-\$1000 Eac Cystal/Resonator High Freq: Startup line: 18C.Cf.+ 64 ms Dol Dol Dol Dol Dol Dol Dol Do
Congen in JT Congen in JT Congen in JT Congen Cocen Sepen Cocen Sepen Essave Boolss Boolss Sup Cocen Sup Sup Cocen Sup Sup Cocen Sup Sup Cocen Sup Sup	AG mode with A Tringg10 AG mode with A Tringg10 Finite LookBin Advanced Hur'Safring Hird Hol Auto Vale Bor Planh scien 1024 words nan address-\$100 Bor Planh scien 1024 words nan address-\$100 Ex Cystal/Records High-Freq; Starup line: 19K.01 - 64 se Dd8 Dd8 Dd8
Chrappan in JT Air Program Fuie OCDEN JTAGEN SPEN EESAVE BOOTRST BOOTRST BOOEN SUT_CKSEL POOEN SUT_CKSEL Auto read Start warnings Start warnings	Add mode with A Transpats Add mode with A Transpats Add mode with A Tran
Reconstruction Reconst	AG mode with X1mega1.2 AG mode with X1mega1.2 Add mode with X1mega1.2
Strappin in JT an Program Pro	AG mode with A Tranga15 AG mode with A Tranga15 Addition to the transmission of the Setting Multiple Auto Value

Rysunek 5. Ustawienie fuse bitów mikrokontrolera za pomocą AVR Studio

płytce są dostępne dwa złącza programujące: JP2 dla programatora JTAG oraz JP1 dla programatora ISP. Jeśli używamy programatora ISP, to zworkę na J3 należy ustawić w pozycji "PROG isp" (zwarte piny 1-2). Przy programowaniu za pomocą JTA-G'a ustawienie J3 nie ma znaczenia.

W materiałach dodatkowych do artykułu znajduje się plik *avr_jtag_m16.hex.hex*. Jest to plik firmware działającego programatora JTAG-ICE. Po podłączeniu zasilania



Rysunek 6. Ustawienie numeru portu COM



Rysunek 7. Numery portów COM w trybie debug AVR Studia

do programatora przez złącze JP3 "JTAG". Po zapisaniu pamięci Flash mikrokontrolera, należy ustawić bity konfiguracyjne zgodnie z **rysunkiem 4** (PonyProg). Dla programatora kompatybilnego z AvrStudio ustawienia będą takie, jak na **rysunku 5**.

Po zaprogramowaniu mikrokontrolera, zworkę J3 przełączamy w pozycję *WORK* (zwarte piny 2-3) lub zdejmujemy ją całkowicie. Jeśli zworka została zdjęta, to bufor U3 jest cały czas dołączony do programowanego systemu. Jeśli zworka zwiera piny 2-3, bufor jest przyłączany tylko na czas programowania. Na J2 zakładamy jumper zwierający piny 1-2 (w pozycji *WORK*). Po wykonaniu powyższych czynności, programator powinien być gotowy do pracy. Jeśli tak nie jest, należy wykonać następujące kroki:

- REKLAMA



Rysunek 9. Numery portów COM w trybie programowania

Urządzenie programujemy plikiem bootloadera (plik *avrboot_(1).hex*), ustawienie bitów konfiguracyjnych jak na rysunku 4 i rysunku 5.

Uruchamiamy program terminala (*HyperTerminal, ExTerm, Termite* itp.) ustawiamy parametry transmisji 19200, 8, N, 1. Numer portu COM odczytujemy z menedżera urządzeń. Musi się ona zawierać w przedziale 1...4. Jeśli wirtualny port COM interfejsu JTAG otrzymał inny numer, należy go zmienić w menedżerze urządzeń (menu podręczne *Właściwości,* zakładka Ustawienia portu, przycisk Zaawansowane – **rysunek 6**).

W programatorze ustawiamy zworkę J2 w pozycji *PROG* (uruchomienie bootloadera), po czym dołączamy zasilanie za pomocą złącza JP3 – *JTAG*. Z terminala wysyłamy znak S. W odpowiedzi powinniśmy otrzymać komunikat *AVRBOOT*.

Z menu Tools środowiska AvrStudio wybieramy AVRprog, a tam wskazujemy na plik upgrade.ebn, który po zainstalowaniu AVR Studia standardowo jest umieszczany w folderze C:\Program Files\ Atmel\AVR Tools\JTAGICE\, po czym naciskamy Program.

Po bezbłędnym załadowaniu programu zworkę J2 przełączamy w pozycję *WORK* (zwarte piny 1-2), natomiast J3 przełączamy zgodnie z wcześniejszym opisem.

Uwaga! W trybie *debug* AvrStudio akceptuje numery portów COM w zakresie 1...4! (**rysunek 7**). W wypadku wgrywania programu do systemu docelowego numery portów COM mogą być zawarte w zakresie 1...9 (**rysunek 8**).

Sławomir Skrzyński, EP

Bibliografia:

Jacek A. Michalski "Interfejs JTAG do procesorów AVR", Elektronika Praktyczna 6/2004, str. 10 (AVT-581)

http://www.m2uu.com/elektronika:avrjtag http://www.elektroda.pl/rtvforum/ topic1463084.html