

# Murasaki Zou

むらさきぞう

v1.1

## Opis programowania modułu LPC2368/LPC1768 z wykorzystaniem ISP

Moduł mikroprocesorowy Murasaki Zou v1.1 wyposażony jest w jeden z dwóch mikrokontrolerów tj. ARM7 LPC2368, oraz Cortex MX LPC1768. Oba mikrokontrolery są wyposażone w interfejs JTAG w normalnych warunkach służący do programowania i debugowania programu.

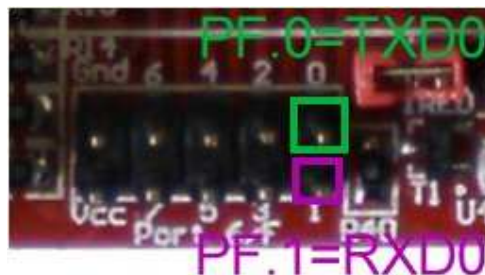
Może jednak zajść potrzeba zaprogramowania modułu metodą ISP czyli In System Programming. Sytuacja taka może być spowodowana zarówno brakiem odpowiedniego „JTAGa” jak i zablokowaniem możliwości programowania przy pomocy tego interfejsu co ma miejsce w wyniku przypadkowego umieszczenia instrukcji blokujących interfejs w pierwszych liniach programu lub w pewnych innych specyficznych przypadkach, w których może dojść do samoczynnego zablokowania (np. po ustawieniu zbyt dużej prędkości komunikacji).

Metoda ISP wykorzystuje bootloader wbudowany w procesor, który komunikuje się z komputerem poprzez interfejs UART0. Do programowania należy w tym wypadku użyć odpowiedniego programu takiego jak Flash Magic lub lpc21isp.

### 1. Przygotowanie makiety

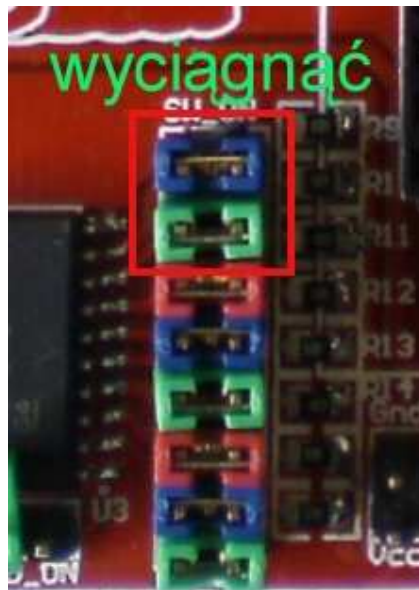
Ponieważ bootloader wykorzystuje UART0 a standardowo do konwertera RS232 lub USB podłączony jest UART1 modułu należy odpowiednio przełączyć te interfejsy. Do wykonania opisanych niżej czynności będą potrzebne trzy przewody obustronnie zakończone konektorami do goldpinów.

Linie RXD0 i TXD0 interfejsu UART0 są połączone z liniami portu F a konkretnie PF.1 i PF.0 na makiecie (rys.1).



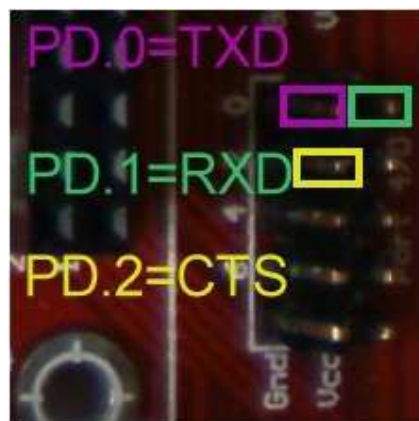
rys.1. Linie TXD0 i RXD0.

Ze względu na specyfikę podłączenia portu F (obsługa przycisków i diod LED na makiecie) należy dodatkowo odłączyć te linie od peryferii makiety wyciągając dwie górne zworki „SW\_ON” (rys.2).



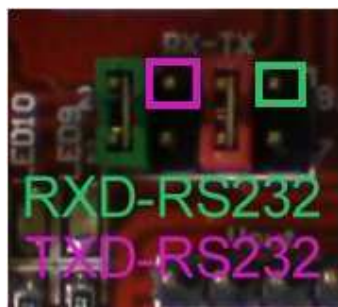
rys.2. Zworki odłączające port F.

Aby przekierować UART0 na odpowiednie linie jednego z konwerterów należy wybrać jeden z wariantów. Po pierwsze można połączyć odpowiadające sobie linie TXD i RXD interfejsów UART0 i UART1 czyli RXD0-RXD1 i TXD0-TXD1. W ten sposób wyboru interfejsu PC (USB lub RS232) dokonuje się standardowo zworkami RX-TX na makiecie. Linie interfejsu UART1 modułu są dostępne na porcie D a konkretnie PD.0 i PD.1 (rys.3).

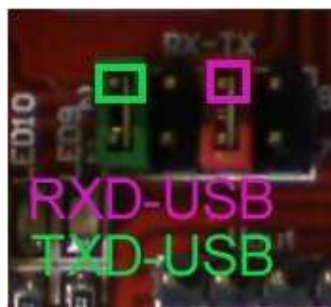


rys.3. Interfejs UART1 na porcie D.

Inną metodą jest dopięcie się bezpośrednio pod konwertery w miejscu zworek RX-TX. Należy wybrać odpowiednie piny zależnie od tego, czy chcemy użyć interfejsu RS232 czy USB co pokazują rysunki 4 i 5 (UWAGA! Opis oznacza miejsce dopięcia odpowiednich linii procesora, dla PC linie RX i TX powinny być nazwane odwrotnie).



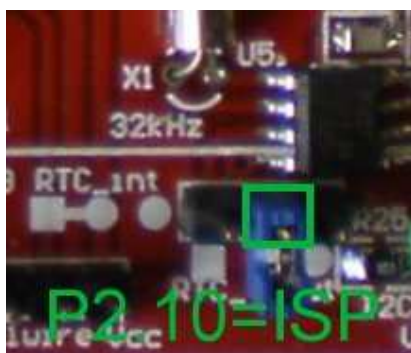
rys.4. Linie RS232.



rys.5. Linie USB.

## 2. Ręczne uruchomienie procesu programowania ISP.

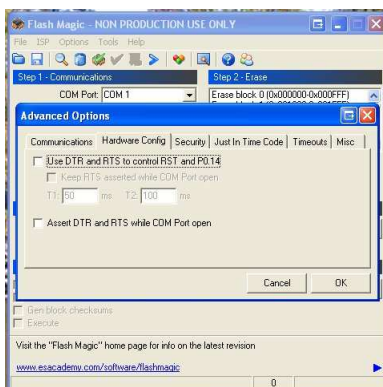
Po wykonaniu czynności opisanych w punkcie pierwszym makieta i moduł są gotowe do programowania w trybie ręcznego uruchomienia procedury ISP. Mechanizm uruchomienia bootloadera wymaga, aby podczas resetu mikrokontrolera linia P2.10 oznaczona ISP była w stanie niskim (logiczne 0, poziom gnd). Linia ta jest dostępna na makiecie jako środkowy pin zworki RTC\_int (rys.6).



rys.6. Linia ISP – P2.10.

**UWAGA!!** W wersji v1.0 linia ISP – P2.10 nie została wyprowadzona. Najprostszym sposobem rozwiązania tego problemu jest fizyczne zwarcie (np. kropelką cyny) linii 53 i 52 mikrokontrolera (P2.10 i P2.11). Linia P2.11 (52) została na module w wersji v1.0 połączona z linią P0.22 (56) wyprowadzoną na makiecie jako PC.7.

W trybie ręcznego wejścia do bootloadera należy linię ISP manualnie na krótki czas dołączyć do masy podczas resetowania procesora przyciskiem reset na makiecie i uruchomić programowanie w programie Flash Magic. Wcześniej należy program skonfigurować tak, aby nie używał linii DTR i RTS do wprowadzania procesora w tryb ISP. Dokonujemy tego wybierając z menu „Options” opcję „Advanced Options”, przechodzimy do zakładki „Hardware Config” i odznaczamy opcję „Use DTR and RTS...” (rys.7).



rys.7. Konfiguracja Flash Magic bez używania linii DTR i RTS.

Chcąc użyć programu lpc21isp (obsługę LPC17XX dodano w wersji z 19.06.2009) należy wywołać go z linii komend w następujący sposób:

*lpc21isp plik\_z\_programem.hex COM BAUDRATE XTAL*

gdzie:

- plik\_z\_programem.hex – plik ze skompilowanym programem dla mikrokontrolera w formacie Intel hex,
- COM – nazwa fizycznego bądź wirtualnego portu COM, do którego podpięto moduł np. com1,
- BAUDRATE – prędkość z jaką będzie programowany mikrokontroler w bps; zaleca się małą prędkość rzędu 38400 bps,
- XTAL – częstotliwość rezonatora kwarcowego, z którego taktowany jest mikrokontroler w kHz np. 12000 dla 12MHz.

### 3. Półautomatyczne programowanie ISP.

Do automatycznego wprowadzenia mikrokontrolera w tryb ISP można wykorzystać linię RTS interfejsu RS232. Linia ta jest dostępna (jako linia CTS procesora) na pinie PD.2 (rys.3) a także na odpowiednim pinie zworek RTS-CTS zależnie od wybranego interfejsu (RS232 lub USB, rys.8).



rys.8. Linia RTS interfejsu RS232 i USB.

Po połączeniu odpowiedniego pinu z linią P2.10 (rys.6) należy skonfigurować program lpc21isp do korzystania z linii DTR i RTS (odwrotnie niż w punkcie 2). Ze względu na konieczność zanegowania linii RTS i brak takiej opcji w programie Flash Magic nie może on być użyty w tym przypadku.

Aby wywołać program lpc21isp z opcjami używania linii RTS i jej negacji należy wykonać polecenie:

*lpc21isp -control -controlinv plik\_z\_programem.hex COM BAUDRATE XTAL*

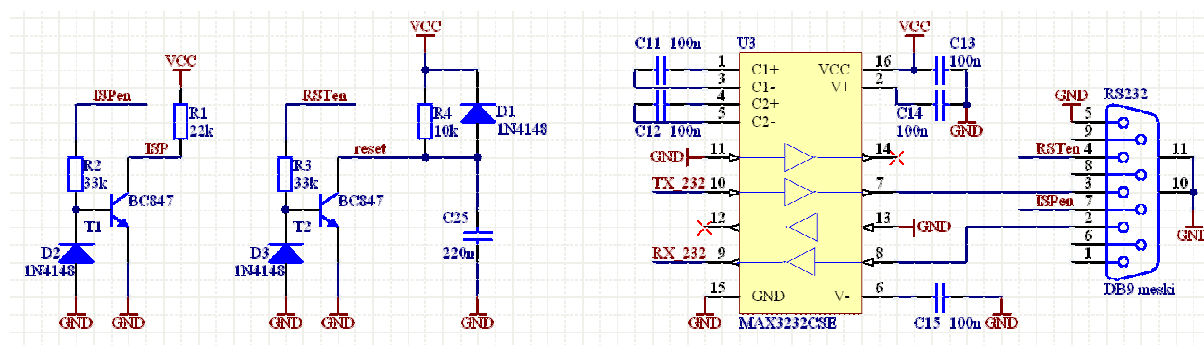
gdzie:

- control – komenda użycia linii DTR i RTS,
- controlinv – komenda negacji linii DTR i RTS.

Po wywołaniu komendy linia ISP zostanie ustawiona w stan niski automatycznie. Należy pamiętać o ręcznym zresetowaniu procesora przyciskiem na makiecie.

#### 4. Automatyczne programowanie ISP.

Zarówno program Flash Magic jak i lpc21isp pozwalają na w pełni automatyczne sterowanie liniami reset i ISP procesora przy wykorzystaniu linii DTR i RTS interfejsu RS232. Schemat klasycznego układu „programatora ISP” pokazano na rysunku 9.



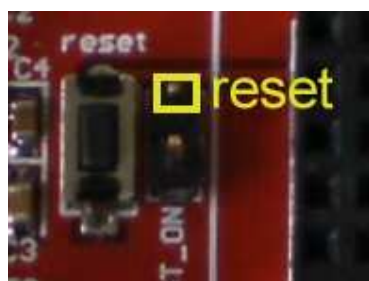
rys.9. Schemat elektryczny „programatora” ISP.

Jak widać poza konwerterem poziomów napięć dla linii RXD i TXD występują tu dwa dodatkowe tranzystory wraz z kilkoma towarzyszącymi im elementami. Tranzystory te pełnią dwie funkcje, po pierwsze następuje tu konwersja napięć do poziomu tolerowanego przez procesor (0-3V3) oraz negacja sygnałów. Elementy D1 i C25 są zbędne, natomiast elementy D2 i D3 pełnią jedynie rolę zabezpieczenia.

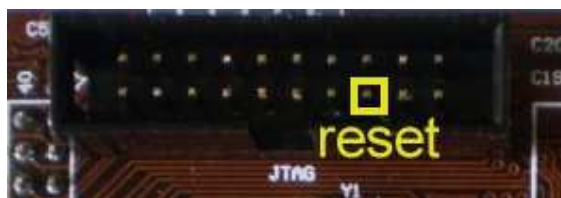
W punkcie 3 opisano sposób wykorzystania linii RTS (nr 7 w złączu DB9) do wprowadzenia mikrokontrolera w stan ISP. Ze względu na wykorzystanie tam jednej z linii konwertera RS232 lub USB, która nie była zanegowana (tak jak czyni to układ tranzystorowy) występowała tam konieczność programowej negacji wspomnianej linii.

Na mackiecie nie przewidziano wykorzystania linii DTR stąd nie jest ona nigdzie wyprowadzona. Chcąc zatem wykorzystać komponenty samej makiety do w pełni automatycznego sterowania procesem programowania ISP należałoby dolutować odpowiedni układ konwersji napięcia do linii nr 4 złącza DB9 na mackiecie. Pamiętać przy tym należy, że układ ten nie może negować sygnału jeżeli ta opcja zostanie uruchomiona w programie lpc21isp (jeżeli dodatkowe komponenty będą negowały stan linii DTR należy zadbać także o negację linii CTS). W najprostszym przypadku układ taki może mieć postać zwykłego rezystorowego dzielnika napięcia lub prostego ogranicznika z rezystorem i diodą Zenera 3V3 (lub kombinacji obu powyższych).

Linia reset procesora dostępna jest na pakiecie obok przycisku reset (górny pin zworki, rys.10), a także na module jako pin nr 15 złącza JTAG (rys 11).



rys.10. Linia reset na mackiecie.



rys.11. Linia reset na module.