

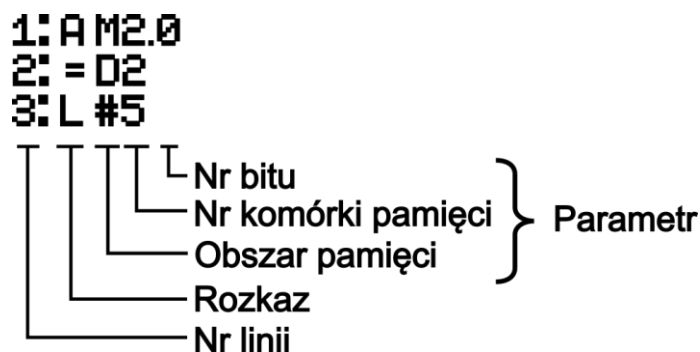
# **Arduino programowane za pomocą wyświetlacza i mini klawiatury**

## Spis treści

1. Szczegółowy opis działania.....	3
Tabela 1. Schemat menu .....	4
Tabela 2. Lista rozkazów.....	5
Tabela 3. Mapa dostępnej pamięci .....	6
Tabela 4. Zarezerwowane obszary pamięci .....	6
2. Przykłady: .....	7
Przykład 1. Rozkaz A :And.....	7
Przykład 2. Rozkaz O :Or.....	7
Przykład 3. Rozkaz AN :Not And .....	7
Przykład 4. Rozkaz ON :Not Or .....	7
Przykład 5. Rozkaz =: Assign .....	7
Przykład 6. Rozkazy S :Set, R :Reset.....	7
Przykład 7. Rozkaz FP :Positive Edge .....	7
Przykład 8. Rozkaz FN :Negative Edge.....	7
Przykład 9. Rozkazy L :Load, T :Transfer .....	7
Przykład 10. Rozkaz SP :Pulse timer .....	8
Przykład 11. Rozkaz SE :Ext. Pulse timer .....	8
Przykład 12. Rozkaz SD :On-Delay timer .....	8
Przykład 13. Rozkaz SS: Ret. On-Delay timer.....	8
Przykład 14. Rozkaz SF :Off-Delay timer.....	8
Przykład 15. Rozkaz R :Reset timer.....	9
Przykład 16. Rozkazy CU :Inc counter, CD :Dec counter, S :Set counter, R :Reset counter, L :Load .	9
Przykład 18. Rozkaz -I : Sub .....	9
Przykład 19. Rozkaz *I : Sub.....	9
Przykład 20. Rozkaz /I : Sub .....	10
Przykład 21. Rozkaz ==I :Equals .....	10
Przykład 22. Rozkaz <>I :Equals .....	10
Przykład 23. Rozkaz >I :Greater .....	10
Przykład 24. Rozkaz <I :Less.....	10
Przykład 25. Rozkaz >= :Greater or equal.....	10
Przykład 26. Rozkaz <= :Less or Equal .....	10
Przykład 27. Rozkazy JU :Jump uncond., JC: Jump cond. ....	11
Przykład 28. Odczyt wartości z wejścia analogowego .....	11
Przykład 29. Zapis wartości do wyjścia analogowego - sterowanie wypełnieniem PWM .....	11

# 1. Szczegółowy opis działania

Istotą działania tego urządzenia jest program wgrany jednorazowo do Arduino. Program ten zajmuje się obsługą klawiatury, wyświetlacza oraz wykonywaniem zadanych przez użytkownika instrukcji. Przy pomocy 5-przyciskowej klawiatury poruszamy się po menu wyświetlanym na wyświetlaczu OLED (schemat menu urządzenia przedstawiony jest w tabeli nr 1). Wybierając odpowiednie pozycje w menu jesteśmy w stanie zaprogramować lub edytować listę instrukcji jakie ma wykonać nasze Arduino. Instrukcje te budową zbliżone są do języka assemblerowego. W poszczególnej instrukcji można wyróżnić następujące elementy:



Zaprogramowane instrukcje wykonują się od góry do dołu (zaczynając od linii nr 1). Do przekazania wyniku kolejnych operacji służy specjalny rejestr RLO (Result od Logic). Wykonanie kolejnych instrukcji może być zależne od stanu tego rejestru, modyfikować go lub anulować jego działanie. Szczegółowy opis wszystkich instrukcji znajduje się w tabeli nr 2.

Parametrem instrukcji może być obszar pamięci bądź wartość liczbowa. Wartości liczbowe parametrów oraz wartości wyników operacji liczbowych zapisywane są w dwóch akumulatorach (ACC – ACC0 i ACC1). Jeśli chodzi o pamięć to mamy do dyspozycji 64 bajty pamięci podzielonej na obszary „MB” – 64 bajty, „MW” – 32 słowa, „MD” – 16 podwójnych słów. Pamięć jest współdzielona pomiędzy wspomnianymi obszarami (Np.: Słowo MW0 składa się z dwóch bajtów MB0 i MB1). Operacje logiczne mają dostęp do poszczególnych bitów obszaru „MB”. Dwa pierwsze bajty pamięci są zarezerwowane (patrz tabela nr 3).

Dodatkowo niektóre instrukcje mają dostęp do wejść/wyjść cyfrowych Arduino (wejścia: D0-D7, wyjścia D8-D13), wejść/wyjść analogowych (wejścia AI0-AI7, wyjścia PWM AO3, AO5, AO6, AO11). Poza tym mamy do dyspozycji 8 timerów 32-bitowych (T0-T7) oraz 8 liczników 32-bitowych (C0-C7).

Mapa dostępnej pamięci znajduje się w tabeli nr 3. Urządzenie przyjmie maksymalnie 64 instrukcji. W celu lepszego zrozumienia systemu najlepiej przeanalizować przykłady znajdujące się w rozdziale nr 2.

Tabela 1. Schemat menu

Poziom 0	Poziom 1	Poziom 2	Poziom 3	Poziom 4	Poziom 5	Poziom 6	Poziom 7
Run	Save&Run	Running...	-	-	-	-	-
	Discard	Running...	-	-	-	-	-
	Cancel	-	-	-	-	-	-
Edit	Listing programu	Tryb edycji	Grupa poleceń	Rozkaz	Parametr	Wartość dla parametru	Wartość dla parametru
	1: ... 2: ... 3: ... [+] (wybór linii)	Insert Edit	Logic	A :And O :Or AN :Not And ON :Not Or	DI:Digital input	Enter input nr:	-
					M:bit marker	Enter byte nr:	Enter bit position:
					T:Timer	Enter timer nr:	-
				= :Assign S :Set R: Reset	M:bit marker	Enter byte nr:	Enter bit position:
					DO:Digital output	Enter output nr:	
				FP: Positive edge FN: Negative Edge	M:bit marker	Enter byte nr:	Enter bit position:
			Move	L :Load	#:Constant	Enter value:	-
					AI:Analog input	Enter input nr:	-
					MB:Memory byte	Enter byte nr:	-
					MW:Memory word	Enter word nr:	-
					MD:Memory double word	Enter dword nr:	-
				T :Transfer	AO:Analog output MB:Memory byte MW:Memory word MD:Memory double word	Enter var position	-
			Timers	SP :Pulse timer SE :Ext. Pulse timer SD :On-Delay timer SS: Ret. On-Delay timer SF :Off-Delay timer R :Reset timer	T:Timer	Enter timer nr:	-
			Counters	CU :Increment counter CD :Decrement counter S :Set counter R :Reset counter L :Load counter	C:Counter	Enter counter nr:	-
			Aritmetic	+I : Add -I :Substract *I :Multiply /I :Divide	-	-	-
			Compare	==I :Equals <>I :Different >I :Greater <I :Less >= :Greater or equal <= :Less or Equal	-	-	-
			Jump	JU :Jump uncond. JC: Jump cond. JCN: Jump cond. If not	@ :Program line number	-	-
		Remove	-	-	-	-	-
Program	Programming eeprom...	-	-	-	-	-	-
Clear	Erasing local program...	-	-	-	-	-	-

Tabela 2. Lista rozkazów

Rozkaz	Opis	Zależny od RLO	Modyfikuje RLO	Anuluje RLO	Odczytuje ACC	Modyfikuje ACC
<b>A :And</b>	Iloczyn logiczny RLO i parametru. Wynik w RLO	TAK	TAK	NIE	NIE	NIE
<b>O :Or</b>	Suma logiczna RLO i parametru. Wynik w RLO	TAK	TAK	NIE	NIE	NIE
<b>AN :Not And</b>	Negacja iloczynu logicznego RLO i parametru. Wynik w RLO	TAK	TAK	NIE	NIE	NIE
<b>ON :Not Or</b>	Negacja sumy logicznej RLO i parametru. Wynik w RLO	TAK	TAK	NIE	NIE	NIE
<b>= :Assign</b>	Przepisanie RLO do parametru	TAK	NIE	TAK	NIE	NIE
<b>S :Set</b>	Ustawienie bitu parametru na 1, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>R: Reset</b>	Reset bitu parametru na 0, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>FP: Positive edge</b>	Ustawienie RLO na 1, gdy RLO zmienia się z 0 na 1	TAK	TAK	NIE	NIE	NIE
<b>FN: Negative Edge</b>	Ustawienie RLO na 1, gdy RLO zmienia się z 1 na 0	TAK	TAK	NIE	NIE	NIE
<b>L :Load</b>	Ładuje wartość z parametru do ACC	NIE	NIE	NIE	NIE	TAK
<b>T :Transfer</b>	Przenosi wartość z ACC do parametru	NIE	NIE	NIE	TAK	NIE
<b>SP :Pulse timer</b>	Wyzwolenie timera impuls, gdy RLO=1 po czasie wczytanym z ACC	TAK	NIE	TAK	TAK	NIE
<b>SE :Ext. Pulse timer</b>	Wyzwolenie timera impuls z pamięcią, gdy RLO=1 po czasie wczytanym z ACC	TAK	NIE	TAK	TAK	NIE
<b>SD :On-Delay timer</b>	Wyzwolenie timera opóźnienie załączenia, gdy RLO=1 po czasie wczytanym z ACC	TAK	NIE	TAK	TAK	NIE
<b>SS: Ret. On-Delay timer</b>	Wyzwolenie timera opóźnienie załączenia z pamięcią, gdy RLO=1 po czasie wczytanym z ACC	TAK	NIE	TAK	TAK	NIE
<b>SF :Off-Delay timer</b>	Wyzwolenie timera opóźnienie wyłączenia, gdy RLO=1 po czasie wczytanym z ACC	TAK	NIE	TAK	TAK	NIE
<b>R :Reset timer</b>	Reset timera	TAK	NIE	TAK	NIE	NIE
<b>CU :Increment counter</b>	Inkrementacja licznika, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>CD :Decrement counter</b>	Dekrementacja licznik, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>S :Set counter</b>	Ustawia stan licznika na ACC, gdy RLO=1	TAK	NIE	TAK	TAK	NIE
<b>R :Reset counter</b>	Zerowanie licznika, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>L :Load counter</b>	Załadowanie wartości licznika do ACC	NIE	NIE	NIE	NIE	TAK
<b>+I : Add</b>	Wykonuje operację: ACC0 = ACC1+ACC0	NIE	NIE	NIE	TAK	TAK
<b>-I :Subtract</b>	Wykonuje operację: ACC0 = ACC1-ACC0	NIE	NIE	NIE	TAK	TAK
<b>*I :Multiply</b>	Wykonuje operację: ACC0 = ACC1*ACC0	NIE	NIE	NIE	TAK	TAK
<b>/I :Divide</b>	Wykonuje operację: ACC0 = ACC1/ACC0	NIE	NIE	NIE	TAK	TAK
<b>==I :Equals</b>	RLO=1, gdy ACC1=ACC0	NIE	TAK	NIE	TAK	NIE
<b>&lt;&gt;I :Different</b>	RLO=1, gdy ACC1<>ACC0	NIE	TAK	NIE	TAK	NIE
<b>&gt;I :Greater</b>	RLO=1, gdy ACC1>ACC0	NIE	TAK	NIE	TAK	NIE
<b>&lt;I :Less</b>	RLO=1, gdy ACC1<ACC0	NIE	TAK	NIE	TAK	NIE
<b>&gt;= :Greater or equal</b>	RLO=1, gdy ACC1>=ACC0	NIE	TAK	NIE	TAK	NIE
<b>&lt;= :Less or Equal</b>	RLO=1, gdy ACC1<=ACC0	NIE	TAK	NIE	TAK	NIE
<b>JU :Jump uncond.</b>	Przenosi wykonanie programu do linii podanej jako parametr	NIE	NIE	NIE	NIE	NIE
<b>JC: Jump cond.</b>	Przenosi wykonanie programu do linii podanej jako parametr, gdy RLO=1	TAK	NIE	TAK	NIE	NIE
<b>JCN: Jump cond. If not</b>	Przenosi wykonanie programu do linii podanej jako parametr, gdy RLO=0	TAK	NIE	TAK	NIE	NIE

Tabela 3. Mapa dostępnej pamięci

Nazwa obszaru	Zakres adresów	Uwagi
Wejścia cyfrowe	D0 ... D7	Piny zgodne z Arduino
Wyjścia cyfrowe	D8 ... D13	Piny zgodne z Arduino
Timery 32-bitowe	T0 ... T7	Najmniejsza jednostka czasu to 10ms
Liczniki 32-bitowe	C0 ... C7	Zakres liczników od -2 147 483,648 do 2 147 483 647
Wejścia analogowe	Ai0 ... Ai7	Piny zgodne z Arduino
Wyjścia analogowe	A00 ... A013	Piny zgodne z Arduino
Markery bitowe	M0.0 ... M64.7	Obszary zajmują tą samą pamięć. Różnią się jedynie dostępem (bit, bajt, słowo, podwójne słowo)
Pamięć (bajty)	MB0 ... MB64	
Pamięć (słowa)	MW0 ... MW32	
Pamięć (podwójne słowa)	MD0 ... MD16	

Tabela 4. Zarezerwowane obszary pamięci

Adres pamięci	Przeznaczenie
M0.0	Zmienia stan co 100ms
M0.1	Zmienia stan co 200ms
M0.2	Zmienia stan co 400ms
M0.3	Zmienia stan co 500ms
M0.4	Zmienia stan co 800ms
M0.5	Zmienia stan co 1000ms
M0.6	Zmienia stan co 1600ms
M0.7	Zmienia stan co 2000ms
M1.0	„firstscan” logiczna jedynka tylko podczas trwania pierwszej pętli programu.
M1.1	Zarezerwowany bit diagnostyczny
M1.2	Zawsze „true”
M1.3	Zawsze „false”
M1.4	Jeśli ustawiony to wyświetlaj na ekranie zawartość pamięci o adresie MW29
M1.5	Jeśli ustawiony to wyświetlaj na ekranie zawartość pamięci o adresie MW30
M1.6	Jeśli ustawiony to wyświetlaj na ekranie zawartość pamięci o adresie MW31
M2.0	Przycisk DÓŁ wciśnięty
M2.1	Przycisk ENTER wciśnięty
M2.2	Przycisk LEWO wciśnięty
M2.3	Przycisk PRAWO wciśnięty
M2.4	Przycisk GÓRA wciśnięty
MW29	Wartość z tego adresu wyświetlana będzie na wyświetlaczu, gdy M1.4 jest ustawiony
MW30	Wartość z tego adresu wyświetlana będzie na wyświetlaczu, gdy M1.5jest ustawiony
MW31	Wartość z tego adresu wyświetlana będzie na wyświetlaczu, gdy M1.6 jest ustawiony

## 2. Przykłady:

### Przykład 1. Rozkaz A :And

Po wciśnięciu przycisku DÓŁ wbudowany LED na płytce Arduino pulsuje z częstotliwością 1Hz.

1	A M0.3	Załaduj stan bitu 3 w bajcie nr 0 (zmieniającego stan do 500ms) do RLO
2	A M2.0	Iloczyn logiczny stanu RLO i stanu przycisku DÓŁ, wynik przepisany do RLO
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 2. Rozkaz O :Or

Wbudowany LED na płytce Arduino pulsuje z częstotliwością 1Hz. Po wciśnięciu przycisku DÓŁ wbudowany LED na płytce Arduino przestaje pulsować i świeci się ciągłym światłem.

1	O M0.3	Załaduj stan bitu 3 w bajcie nr 0 (zmieniającego stan do 500ms) do RLO
2	O M2.0	Suma logiczna stanu RLO i stanu przycisku DÓŁ, wynik przepisany do RLO
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 3. Rozkaz AN :Not And

LED na płytce Arduino pulsuje z częstotliwością 1Hz. Po wciśnięciu przycisku DÓŁ LED gaśnie.

1	AN M0.3	Załaduj negację stanu bitu 3 w bajcie nr 0 (zmieniającego stan do 500ms) do RLO
2	AN M2.0	Iloczyn logiczny stanu RLO i negacji stanu przycisku DÓŁ, wynik przepisany do RLO
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 4. Rozkaz ON :Not Or

Wbudowany LED na płytce Arduino świeci ciągłym światłem. Po wciśnięciu przycisku DÓŁ wbudowany LED na płytce Arduino pulsuje z częstotliwością 1Hz.

1	ON M0.3	Załaduj stan bitu 3 w bajcie nr 0 (zmieniającego stan do 500ms) do RLO
2	ON M2.0	Suma logiczna stanu RLO i negacji stanu przycisku DÓŁ, wynik przepisany do RLO
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 5. Rozkaz =: Assign

Wbudowany LED na płytce Arduino świeci ciągłym światłem.

1	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
2	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 6. Rozkazy S :Set, R :Reset

Po wciśnięciu przycisku DÓŁ LED na płytce Arduino zapala się. Po wciśnięciu przycisku GÓRA LED na płytce Arduino gaśnie.

1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	S D13	Jeśli RLO=1 ustaw wyjście cyfrowe nr 13 (wbudowany LED na płytce Arduino)
3	A M2.4	Załaduj stan bitu 4 w bajcie nr 2 (przycisku GÓRA) do RLO
4	R D13	Jeśli RLO=1 zresetuj wyjście cyfrowe nr 13 (wbudowany LED na płytce Arduino)

### Przykład 7. Rozkaz FP :Positive Edge

Po wciśnięciu przycisku DÓŁ LED na płytce Arduino zapala się na krótką chwilę (program wykrywa zbocze wzrastające – zmianę stanu przycisku z 0 na 1).

1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	FP M3.0	Jeśli RLO zmienia stan z 0 na 1 ustaw RLO=1
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 8. Rozkaz FN :Negative Edge

Po wciśnięciu i puszczeniu przycisku DÓŁ LED na płytce Arduino zapala się na krótką chwilę (program wykrywa zbocze wzrastające – zmianę stanu przycisku z 1 na 0).

1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	FN M3.0	Jeśli RLO zmienia stan z 1 na 0 ustaw RLO=1
3	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

### Przykład 9. Rozkazy L :Load, T :Transfer

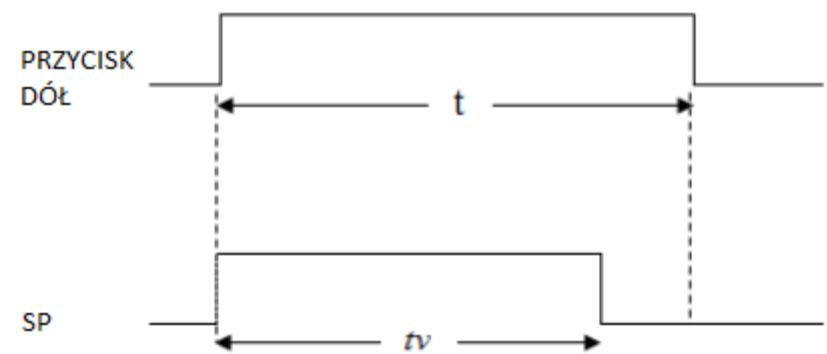
Załadowanie wartości stałej do komórki pamięci.

1	L #100	Załaduj do akumulatora wartość 100
2	T MW2	Załaduj wartość z akumulatora do słowa nr 2

Przykład 10. Rozkaz SP :Pulse timer

Po przyciśnięciu i przytrzymaniu przycisku DÓŁ dłużej niż 2000ms LED na płytce Arduino zapala się na czas 2000ms. W razie puszczenia przycisku DÓŁ wcześniej niż po upływie 2000ms LED na płytce Arduino gaśnie.

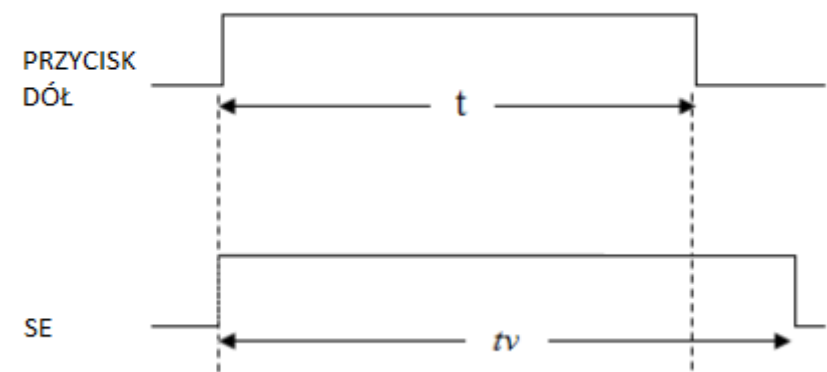
1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SP T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)



Przykład 11. Rozkaz SE :Ext. Pulse timer

Po przyciśnięciu i puszczeniu przycisku DÓŁ LED na płytce Arduino zapala się na czas 2000ms. W razie puszczenia przycisku DÓŁ wcześniej niż po upływie 2000ms LED na płytce Arduino nie gaśnie. Przytrzymanie przycisku DÓŁ na czas dłuższy jak 2000ms powoduje ponowne wyzwolenie timera.

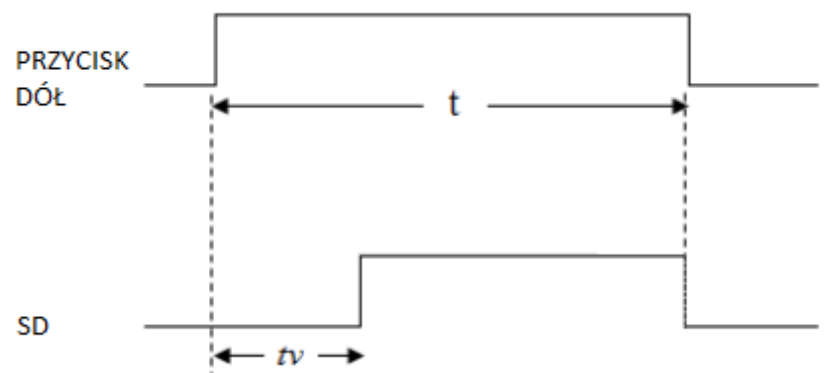
1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SE T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)



Przykład 12. Rozkaz SD :On-Delay timer

Po przyciśnięciu i puszczeniu przycisku DÓŁ LED na płytce Arduino zapala się po czasie 2000ms. W razie puszczenia przycisku DÓŁ wcześniej niż po upływie 2000ms czas timer zeruje się.

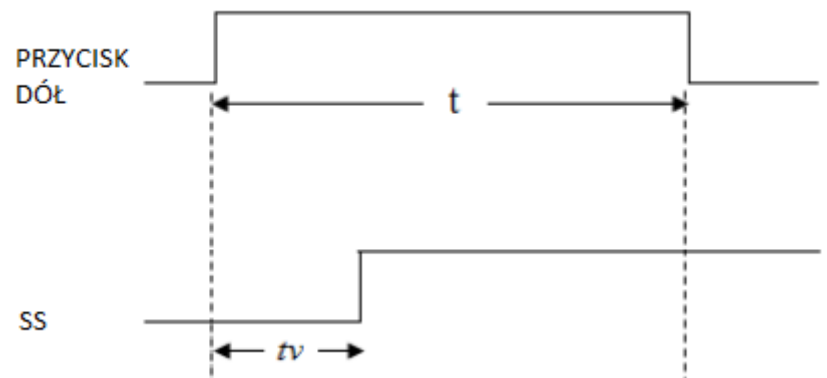
1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SD T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)



Przykład 13. Rozkaz SS: Ret. On-Delay timer

Po przyciśnięciu i puszczeniu przycisku DÓŁ LED na płytce Arduino zapala się po czasie 2000ms. W razie puszczenia przycisku DÓŁ wcześniej niż po upływie 2000ms czas timer nie zeruje się.

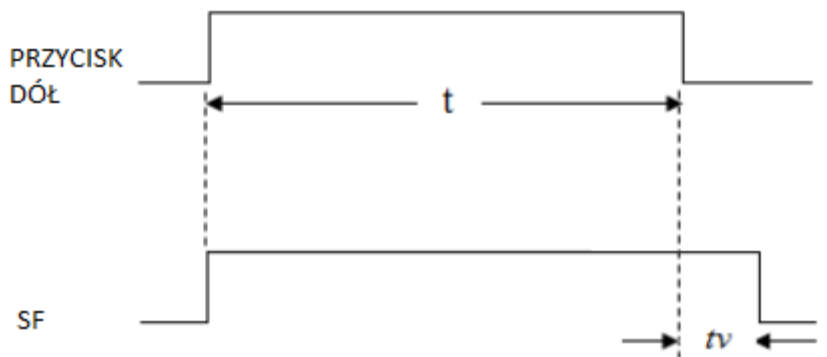
1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SS T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)



Przykład 14. Rozkaz SF :Off-Delay timer

Po przyciśnięciu przycisku DÓŁ LED na płytce Arduino zapala się. W razie puszczenia przycisku DÓŁ LED na płytce Arduino gaśnie po upływie 2000ms.

1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SF T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

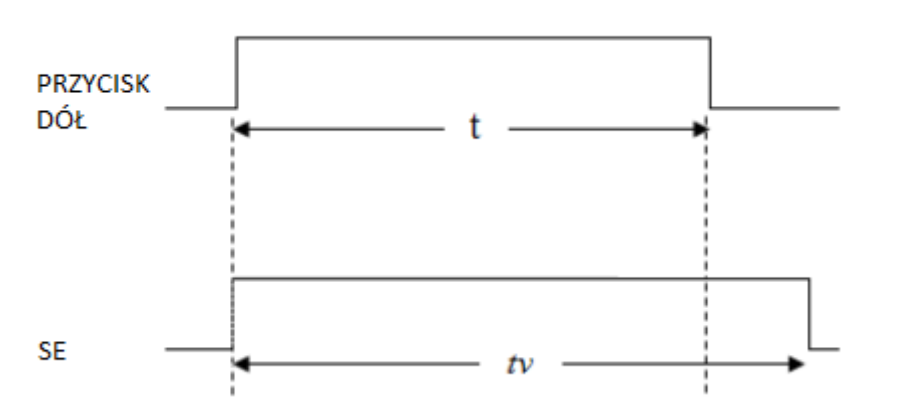




Przykład 15. Rozkaz R :Reset timer

Po przyciśnięciu i puszczeniu przycisku DÓŁ LED na płytce Arduino zapala się na czas 2000ms. W razie puszczenia przycisku DÓŁ wcześniej niż po upływie 2000ms LED na płytce Arduino nie gaśnie. W razie przyciśnięcia przycisku GÓRA wcześniej niż po upływie 2000ms LED na płytce Arduino gaśnie. Przytrzymanie przycisku DÓŁ na czas dłuższy jak 2000ms powoduje ponowne wyzwolenie timera.

1	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
2	L #2000	Załaduj do akumulatora wartość 2000 (tv)
3	SE T0	Wyzwól timer nr 0 jeśli RLO=1 (wartość czasu w ms odczytana z akumulatora)
4	A T0	Załaduj stan timer 0 do RLO
5	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)
6	A M2.4	Załaduj stan bitu 0 w bajcie nr 2 (przycisku GÓRA) do RLO
7	R T0	Resetuj timer 0



Przykład 16. Rozkazy CU :Inc counter, CD :Dec counter, S :Set counter, R :Reset counter, L :Load

Wyświetlacz pokazuje wartość 0. Po wciśnięciu przycisku GÓRA wartość wzrasta. Po wciśnięciu przycisku DÓŁ wartość maleje. Po wciśnięciu przycisku ENTER wartość zostaje wyzerowana. Po wciśnięciu przycisku PRAWO wartość zostaje ustawiona na 10.

1	A M2.4	Załaduj stan bitu 4 w bajcie nr 2 (przycisku GÓRA) do RLO
2	FP M3.0	Jeśli RLO zmienia stan z 0 na 1 ustaw RLO=1
3	CU C0	Jeśli RLO=1 zwiększ wartość licznika C0 o 1
4	A M2.0	Załaduj stan bitu 0 w bajcie nr 2 (przycisku DÓŁ) do RLO
5	FP M3.1	Jeśli RLO zmienia stan z 0 na 1 ustaw RLO=1
6	CD C0	Jeśli RLO=1 zmniejsz wartość licznika C0 o 1
7	A M2.1	Załaduj stan bitu 1 w bajcie nr 2 (przycisku ENTER) do RLO
8	R C0	Jeśli RLO=1 resetuj wartość licznika C0 na 0
9	L #10	Załaduj do akumulatora wartość 10
10	A M2.3	Załaduj stan bitu 3 w bajcie nr 2 (przycisku LEWO) do RLO
11	S C0	Jeśli RLO=1 ustaw wartość licznika C0 na wartość z akumulatora
12	L C0	Załaduj do akumulatora wartość licznika C0
13	T MW29	Załaduj z wartość z akumulatora do słowa nr 29
14	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
15	= M1.4	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)

Przykład 17. Rozkaz +I : Add

Wyświetla sumę dwóch liczb 2 i 3.

1	L #2	Załaduj do akumulatora wartość 2
2	L #3	Załaduj do akumulatora wartość 3
3	+I	Dodaj wartości załadowane do akumulatora. Wynik pozostaje w akumulatorze
4	T MW29	Prześlij wartość z akumulatora do słowa nr 29
5	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
6	= M1.4	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)

Przykład 18. Rozkaz -I : Sub

Wyświetla różnicę dwóch liczb 2 i 3.

1	L #2	Załaduj do akumulatora wartość 2
2	L #3	Załaduj do akumulatora wartość 3
3	-I	Odejmij wartości załadowane do akumulatora. Wynik pozostaje w akumulatorze
4	T MW29	Prześlij wartość z akumulatora do słowa nr 29
5	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
6	= M1.4	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)

Przykład 19. Rozkaz \*I : Sub

Wyświetla iloczyn dwóch liczb 2 i 3.

1	L #2	Załaduj do akumulatora wartość 2
2	L #3	Załaduj do akumulatora wartość 3
3	*I	Przemnoż wartości załadowane do akumulatora. Wynik pozostaje w akumulatorze
4	T MW29	Prześlij wartość z akumulatora do słowa nr 29
5	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
6	= M1.4	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)

Przykład 20. Rozkaz /I : Sub  
Wyświetla iloraz dwóch liczb 10 i 3.

1	L #10	Załaduj do akumulatora wartość 10
2	L #3	Załaduj do akumulatora wartość 3
3	/I	Podziel wartości załadowane do akumulatora. Wynik pozostaje w akumulatorze
4	T MW29	Prześlij wartość z akumulatora do słowa nr 29
5	A M1.2	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „True”) do RLO
6	= M1.4	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)

Przykład 21. Rozkaz ==I :Equals  
LED na płytce Arduino zapala się, gdy porównywane wartość z akumulatora są równe.

1	L #10	Załaduj do akumulatora wartość 10
2	L #10	Załaduj do akumulatora wartość 3
3	==I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 22. Rozkaz <>I :Equals  
LED na płytce Arduino zapala się, gdy porównywane wartość z akumulatora są różne.

1	L #10	Załaduj do akumulatora wartość 10
2	L #11	Załaduj do akumulatora wartość 11
3	<>I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 23. Rozkaz >I :Greater  
LED na płytce Arduino zapala się, gdy pierwsza podana wartość w akumulatorze jest większa od drugiej.

1	L #10	Załaduj do akumulatora wartość 10
2	L #11	Załaduj do akumulatora wartość 11
3	>I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 24. Rozkaz <I :Less  
LED na płytce Arduino zapala się, gdy pierwsza podana wartość w akumulatorze jest mniejsza od drugiej.

1	L #11	Załaduj do akumulatora wartość 11
2	L #10	Załaduj do akumulatora wartość 10
3	<I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 25. Rozkaz >= :Greater or equal  
LED na płytce Arduino zapala się, gdy pierwsza podana wartość w akumulatorze jest większa bądź równa drugiej wartości.

1	L #12	Załaduj do akumulatora wartość 12
2	L #10	Załaduj do akumulatora wartość 10
3	>=I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 26. Rozkaz <= :Less or Equal  
LED na płytce Arduino zapala się, gdy pierwsza podana wartość w akumulatorze jest mniejsza bądź równa drugiej wartości.

1	L #10	Załaduj do akumulatora wartość 11
2	L #10	Załaduj do akumulatora wartość 10
3	<=I	Porównaj wartości załadowane do akumulatora. Wynik w RLO
4	= D13	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 27. Rozkazy JU :Jump uncond., JC: Jump cond.

Program co sekundę zwiększa wartość licznika C0 (wyświetla wartość na wyświetlaczu). Gdy wartość licznika przekroczy 15 następuje wyzerowanie licznika.

1	<b>A M0.3</b>	Załaduj stan bitu 3 w bajcie nr 0 (zmienia stan co 500ms) do RLO
2	<b>FP M3.0</b>	Jeśli RLO zmienia stan z 0 na 1 ustaw RLO=1
3	<b>CU C0</b>	Jeśli RLO=1 zwiększ wartość licznika C0 o 1
4	<b>L C0</b>	Załaduj do akumulatora wartość licznika C0
5	<b>T MW29</b>	Załaduj wartość akumulatora do słowa nr 29
6	<b>A M1.2</b>	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „TRUE”) do RLO
7	<b>= M1.4</b>	Przypisz wartość z RLO do bitu 4 w bajcie nr 1 (Wyświetlaj na ekranie zawartość pamięci o adresie MW29)
8	<b>L C0</b>	Załaduj do akumulatora wartość licznika C0
9	<b>L #15</b>	Załaduj do akumulatora wartość 15
10	<b>&gt;I</b>	Porównaj wartości załadowane do akumulatora (wynik RLO=1 jeśli C0>15)
11	<b>JC @13</b>	Jeśli wartość licznika C0 jest większa niż 15 przenieś wykonanie programu do linii nr 13
12	<b>JU @1</b>	Przenieś wykonanie programu do linii nr 1
13	<b>A M1.2</b>	Załaduj stan bitu 2 w bajcie nr 1 (zawsze „TRUE”) do RLO
14	<b>L #0</b>	Załaduj do akumulatora wartość 0
15	<b>S C0</b>	Jeśli RLO=1 ustaw wartość licznika C0 na wartość z akumulatora

Przykład 28. Odczyt wartości z wejścia analogowego

LED na płytce Arduino zapala się, gdy pierwsza podana wartość wejścia analogowego nr 0 jest większa od 100 (wartości analogowe odczytywane są do zmiennych 8 bitowych – zakres od 0 do 255).

1	<b>L AI0</b>	Załaduj do akumulatora wartość wejścia analogowego AI0
2	<b>L #100</b>	Załaduj do akumulatora wartość 100
3	<b>&gt;I</b>	Porównaj wartości załadowane do akumulatora. (wynik RLO=1 jeśli AI0>100)
4	<b>= D13</b>	Przepisz RLO do wyjścia cyfrowego nr 13 (wbudowany LED na płytce Arduino)

Przykład 29. Zapis wartości do wyjścia analogowego - sterowanie wypełnieniem PWM

LED podłączony do wyjścia A11 powoli rozjaśnia się i powoli przygasa.

1	<b>AN M3.0</b>	Załaduj negację stanu bitu 0 w bajcie nr 3
2	<b>CU C0</b>	Jeśli RLO=1 zwiększ wartość licznika C0 o 1
3	<b>A M3.0</b>	Załaduj stan bitu 0 w bajcie nr 3
4	<b>CD C0</b>	Jeśli RLO=1 zmniejsz wartość licznika C0 o 1
5	<b>L C0</b>	Załaduj do akumulatora wartość licznika C0
6	<b>L #255</b>	Załaduj do akumulatora wartość 255
7	<b>==I</b>	Porównaj wartości załadowane do akumulatora (wynik RLO=1 jeśli równe)
8	<b>S M3.0</b>	Jeśli RLO=1 ustaw na „1” bit 0 w bajcie nr 3
9	<b>L C0</b>	Załaduj do akumulatora wartość licznika C0
10	<b>L #0</b>	Załaduj do akumulatora wartość 0
11	<b>==I</b>	Porównaj wartości załadowane do akumulatora (wynik RLO=1 jeśli równe)
12	<b>R M3.0</b>	Jeśli RLO=1 ustaw na „0” bit 0 w bajcie nr 3
13	<b>L C0</b>	Załaduj do akumulatora wartość licznika C0
14	<b>T A11</b>	Przenieś wartość z akumulatora do wyjścia analogowego A11