

# Opis Symulatora i Wizualizatora Linii Montażowej

Na potrzeby Instytutu Automatyki i Informatyki Stosowanej

Patryk Chaber

19 kwietnia 2020

## 1 Symulator

Przedstawiony dalej symulator zrealizowany jest jako 3 powiązane ze sobą podprogramy działające na (symulowanym) PLC w trzech różnych trybach:

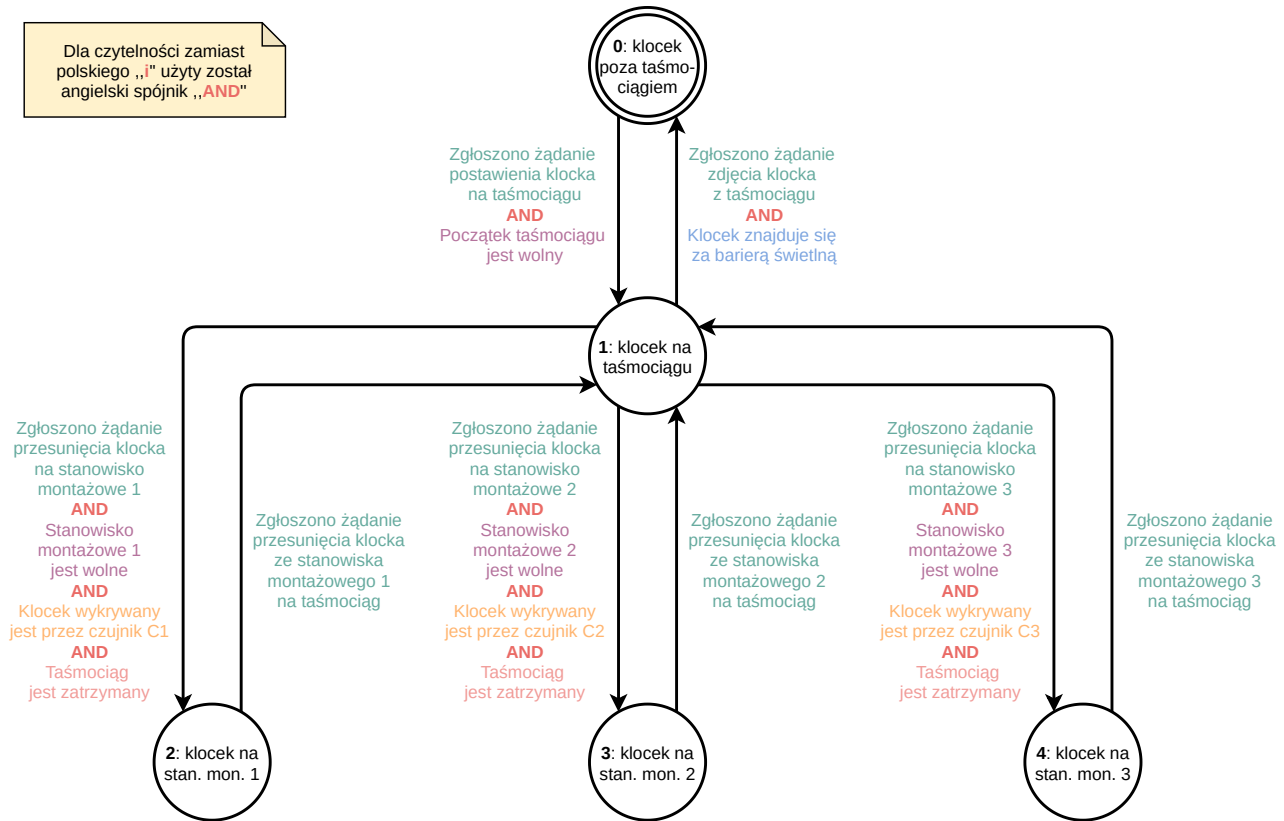
- **LMInit (Initial)**: jest to program zapewniający poprawne wartości zmiennych wykorzystywanych przez pozostałe podprogramy,
- **LMSim\_quick (Scan)**: jest to program przepisujący informację o stanie przycisków pozyskaną z wizualizatora do faktycznych wejść sterownika (jest to rozwiązanie problemu zacinających się wejść),
- **LMSim (Fixed Scan)**: jest to zaimplementowany automat stanów reprezentujący linię montażową, aktualizowany jest co **pół sekundy**.

O ile kod dwóch pierwszych podprogramów jest krótki i dość oczywisty, o tyle prezentowany dalej automat stanów procesu sumulowanego nie jest trywialny.

Przyjęte zostały następujące założenia:

- taśmociąg porusza się tylko w prawą stronę,
- można wstawić jednocześnie do dwóch klocków na taśmociąg,
- przesunięcia klocków (wykonywane w rzeczywistym stanowisku ręcznie) reprezentowane są przez poszczególne przyciski (z ograniczeniami, o których dalej),
- przyciski reprezentujące przesunięcia ręczne zrealizowane są jako bistabilne i powinny być interpretowane jako „zgłoszenie żądania przesunięcia klocka z ... do ...”; zmiana stanu przycisku z wciśniętego na wyciśnięty oznacza, że żądanie zostało spełnione przez symulator (ręczne „wyciśnięcie” przycisku oznacza rezygnację z żądania),
- klocki reprezentowane są jako obiekty przesuwające się wzdłuż taśmociągu niezależnie – oznacza to, że każdy z klocków opisany jest jego pozycją na taśmociągu i to z niej wynika informacja o zapaleniu czujnika lub jego zgaszeniu,
- użytkownik nie będzie mógł wykonać akcji przesunięcia jeśli jego bezpośrednim następstwem będzie kolizja,
- w przypadku wystąpienia kolizji, które nie można było zapobiec (np. użytkownik postawił dwa klocki na taśmociągu i pozwolił im dojechać do skrajnie prawej ściany), wyświetlona zostanie informacja o wystąpieniu kolizji – jest to błąd nienaprawialny (równoznaczny z uszkodzeniem materiału i/lub sprzętu) i jest niedopuszczalny.

Dla czytelności zamiast polskiego „i” użyty został angielski spójnik „AND”



Każdy z klocków reprezentowany jest jako osobne istnienie, któremu przypisany jest jego automat stanu (graficznie przedstawiony powyżej). Każdy klocek więc może znajdować się w stanach (nie zostały im nadane skrótowe nazwy):

- 0: poza stanowiskiem (oczekuje na położenie na taśmociąg),
- 1: na taśmociągu,
- 2: na stacji montażowej 1,
- 3: na stacji montażowej 2,
- 4: na stacji montażowej 3.

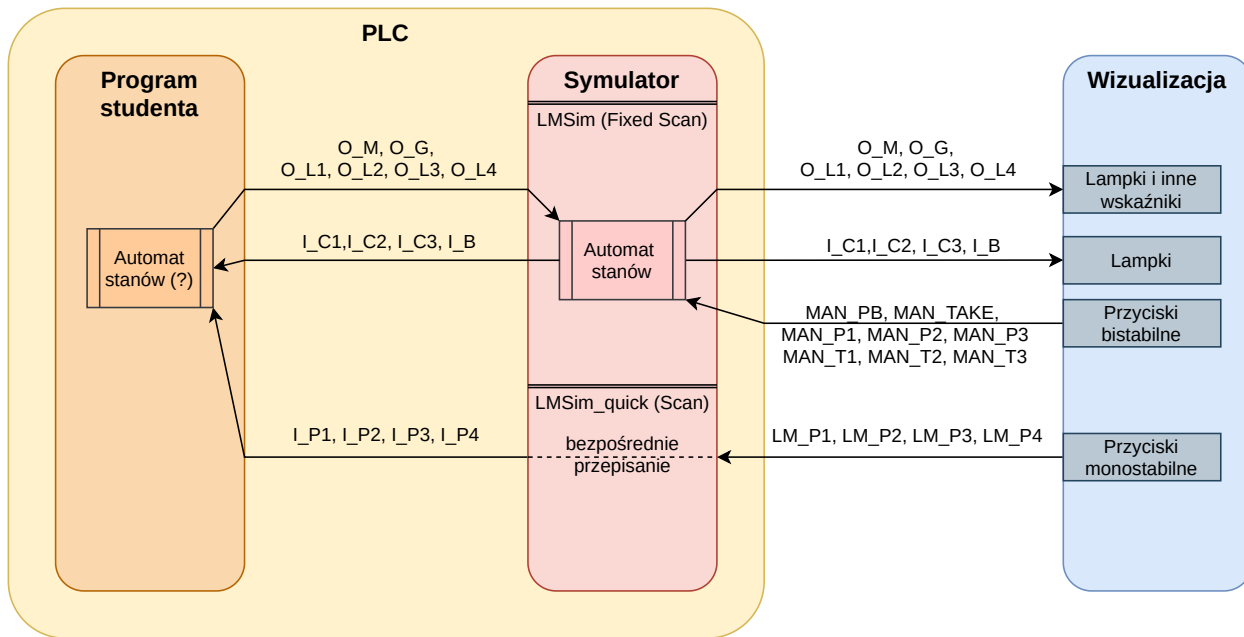
Nie został wydzielony osobny stan reprezentujący bycie na skrajnie prawej pozycji – wynika to z pozycji klocka. Przejście pomiędzy kolejnymi stanami jest zgodne z intuicją, tj.:

- ze stanu 0 można przejść wyłącznie do stanu 1, wyłącznie poprzez ręczne postawienie klocka na taśmociąg i pod warunkiem, że początek taśmociągu jest wolny,
- ze stanu 1 klocek przechodzi do stanu 2,3,4 wyłącznie jeśli spełnione są warunki:
  - klocek znajduje się przy czujniku 1,2,3,
  - wciśnięto przycisk przesunięcia klocka na stanowisko 1,2,3,
  - stanowisko 1,2,3 jest wolne,
  - taśmociąg jest zatrzymany,
- ze stanu 2,3,4 do stanu 1 klocek przechodzi pod wpływem wciśnięcia przycisku przesunięcia klocka ze stanowiska 1,2,3 na taśmociąg bez względu na obecną sytuację na taśmociągu (klocek wraca na miejsce na taśmociągu z którego został zabrany),
- ze stanu 1 klocek przechodzi do stanu 0 tylko i wyłącznie gdy znajduje się w pełni za barierą świetlną i zostanie wciśnięty przycisk zdjęcia klocka z taśmy.

W przypadku gdy któryś z warunków nie zostanie spełniony – nie następuje zmiana stanu.

Warto zauważyć, że symulator nie wymaga wizualizatora do poprawnego działania – ręczne przesuwanie klocka może być symulowane poprzez zmianę sygnałów z tym związanych (tj. ustawianie sygnałów odpowiadających zgłoszeniom kolejnych przesunięć klocka).

Przesył informacji (wartości sygnałów binarnych) między programem studenta, symulatorem a wizualizacją przedstawiony jest na poniższej grafice.



O\_XX (wyjścia):  
M -- silnik,  
G -- głośnik  
L# -- lampka #

I\_XX (wejścia):  
B -- bariera świetlna  
C# -- czujnik nr #  
P# -- przycisk #

LM\_XX (wejścia z wizualizacji):  
P# -- przycisk #

MAN\_XX (przesuwanie ręczne):  
PB -- położenie klocka na taśmę  
TAKE -- zabranie klocka z taśmy  
P# -- położenie klocka na stanowisko #  
T# -- zabranie klocka ze stanowiska #

Na grafice tej pominięte zostały informacje służące wyłącznie wizualizacji, jak np. pozycja klocka, informacja o błędzie. Są one nieistotne z punktu widzenia studenta. Wyraźnie widoczne jest, że student korzysta wyłącznie z sygnałów, które wynikają z oryginalnej definicji stanowiska linii montażowej. W szczególności nie korzysta on z wartości sygnałów reprezentujących przesuwanie ręczne, a także nie wpływa/nie korzysta z informacji dodatkowych o klockach (np. o ich pozycji). Dokładniej: mimo, że praktycznie student ma możliwość odczytu i modyfikowania wartości wszystkich sygnałów, to **student powinien swoje działania ograniczyć do sygnałów I\_, O\_ oraz tych zdefiniowanych przez siebie.**

## 2 Zgodność z treścią „Podstawy sterowania. Ćwiczenia laboratoryjne”

Udostępniony symulator jest w pełni zgodny z istniejącym rzeczywiście stanowiskiem o nazwie Linia Montażowa. Oznacza to, że program działający na symulatorze będzie działać również bez żadnych modyfikacji na prawdziwym stanowisku. Rzeczywista Linia Montażowa oryginalnie jest podłączona do sterownika firmy Siemens, natomiast symulator napisany jest pod kątem wykorzystania sterownika Mitsubishi, stąd należy przyjąć nową (analogiczną) adresację.

Tabela adresów sygnałów wejściowych i wyjściowych (zgodnych ze sterownikami Mitsubishi), ich znaczenia (zgodne z treścią w skrypcie) i ich etykiety w programie przedstawione zostały poniżej:

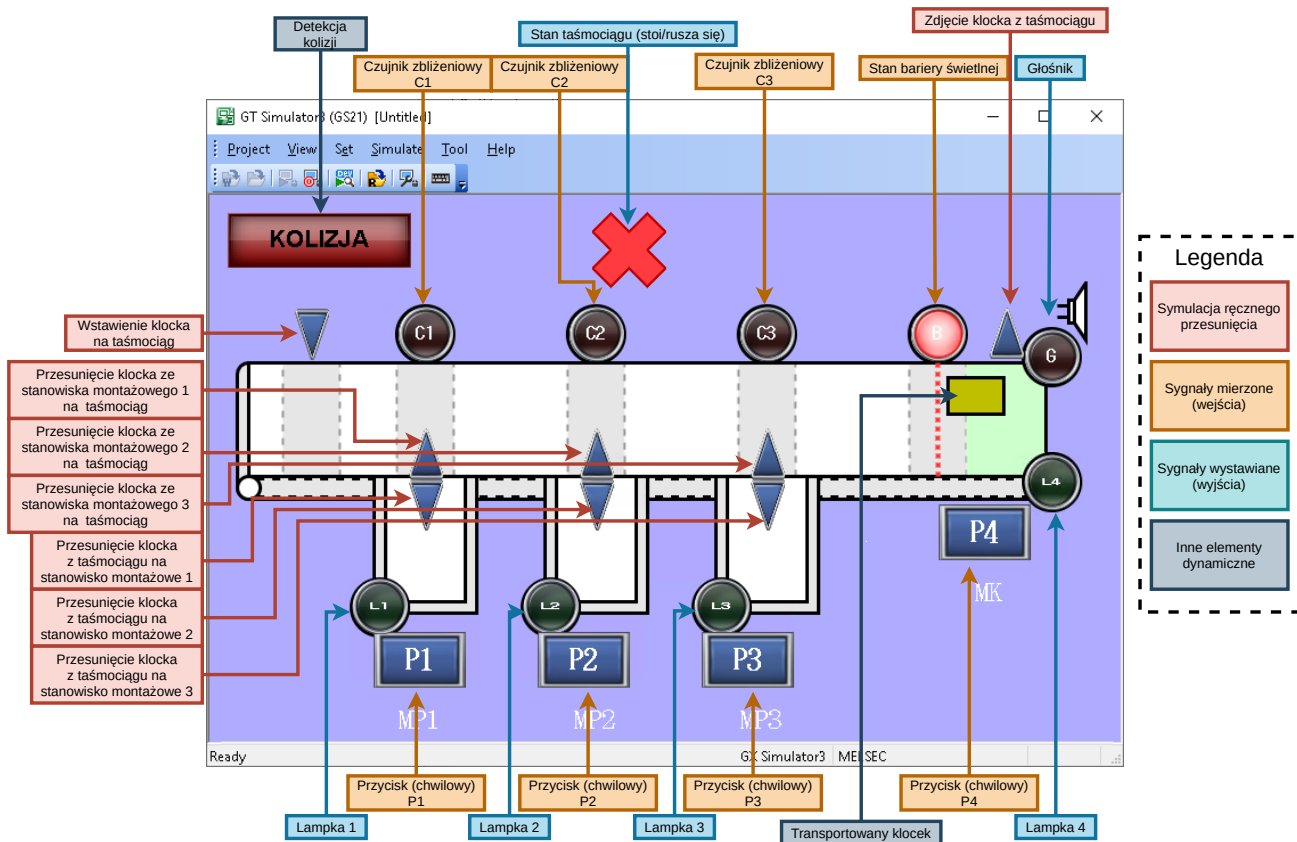
Numer bitu	X7	X6	X5	X4	X3	X2	X1	X0
Znaczenie	<i>b</i>	<i>c3</i>	<i>c2</i>	<i>c1</i>	<i>p4</i>	<i>p3</i>	<i>p2</i>	<i>p1</i>
Etykieta w programie	I_B	I_C3	I_C2	I_C1	I_P4	I_P3	I_P2	I_P1

Numer bitu	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Znaczenie	<i>g</i>	<i>m2</i>	<i>m1</i>	<i>l4</i>	<i>l3</i>	<i>l2</i>	<i>l1</i>
Etykieta w programie	O_G	—	O_M	O_L4	O_L3	O_L2	O_L1

Ze względu na praktyczny brak zastosowania sygnału *m2* nie został on uwzględniony w udostępnionym oprogramowaniu – należy założyć, że jego wartość jest stale równa 0.

## 3 Wizualizacja

Wizualizacja jest inspirowana oryginalnym schematem linii montażowej, z książki „Podstawy sterowania. Ćwiczenia Laboratoryjne” znajdującej się w laboratorium Automatyki. Poniżej przedstawiony jest widok wizualizacji wraz z oznaczeniem co oznaczają poszczególne jej elementy.



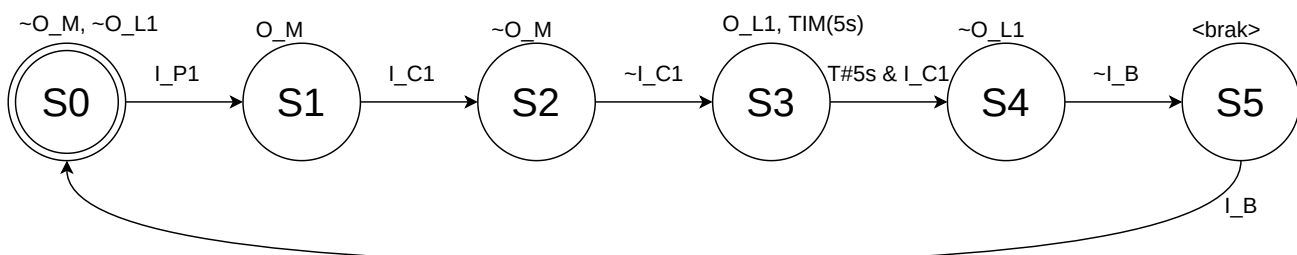
**UWAGA:** symulacja niestety jest dość powolna bez względu na zastosowane usprawnienia po stronie sterownika programowalnego. W związku z tym uprasza się o nieprzesadnie szybkie wciskanie kolejnych przycisków. Oprogramowanie jest stabilne bez względu na tempo klikania, lecz w takiej sytuacji wizualizator wyraźnie zwalnia, co może powodować błędną interpretację stanu stanowiska przez studenta. Częstotliwość przełączania sygnałów na wizualizacji może nie być precyzyjną reprezentacją tego, co dzieje się po stronie sterownika. Dlatego też do sprawdzenia czy np. lampki przełączają się z poprawną częstotliwością należy dokonać korzystając z podglądu programu GxWorks, a nie z wizualizatora.

## 4 Przykładowy program studenta

Na potrzeby prezentacji działania przedstawionego rozwiązania zaprojektowany i zaimplementowany został przykładowy trywialny projekt wykorzystujący podstawowe własności stanowiska. Rozważane zadanie polega na realizacji następujących czynności:

1. Po wciśnięciu przycisku P1 należy rozpocząć transport klocka do stanowiska 1.
2. W momencie przestawienia klocka na stanowisko 1 należy zapalić lampkę L1 – sygnalizuje to zajętość stanowiska.
3. Po upływie 5 sekund i odstawieniu klocka na taśmociąg należy kontynuować jego transport do stanowiska odbiorczego, aż klocek znajdzie się za barierą świetlną.
4. Po zakończeniu transportu należy powrócić do oczekiwania na wciśnięcie przycisku P1 (punkt 1).

Program został napisany w języku LAD. Problem wymaga użycia opóźnienia sygnału za pomocą bloku funkcyjnego TON (do tego podejścia zachęca się studentów). Rozwiązanie zostało zaimplementowane jako automat stanu w postaci widocznej na następującym rysunku:



Oznaczenie  $\sim$  oznacza negację sygnału, natomiast **T#5s** oznacza okres pięciu sekund (notacja zaczerpnięta z oprogramowania GxWorks). Dla przejrzystości zamiast polskiego słowa „i” jako znak logiczny koniunkcji został użyty znak  $\&$ .

## 5 Zgłaszanie problemów

W przypadku problemów z symulatorem lub wizualizacją proszę o kontakt na adres [pjchaber@gmail.com](mailto:pjchaber@gmail.com) lub (preferowane) poprzez platformę Microsoft Teams ([patryk.chaber@pw.edu.pl](mailto:patryk.chaber@pw.edu.pl)).