Projekt linii produkcyjnej z wykorzystaniem sterownika PLC

AiR III rok, semestr 5

Spis treści

Część 1

Ogólny opis algorytmu

1.1 Założenia

Idea algorytmu:

- **Zarządzanie paczkami** - w pierwszej kolejności paczki są rozpoznawane a następnie na podstawie ich wielkości przypisywana im jest trasa narzucona uprzednio przez użytkownika. Każda z taśm ma przypisaną do siebie kolejkę (FIFO), dzięki temu na bieżąco można śledzić dokąd powinna pojechać aktualnie 'przetwarzana' paczka. Na podstawie tej informacji operują zwrotnice, które przekazują paczkę na kolejną linię bądź umożliwiają jej spadnięcie do strefy docelowej.
- **Zarządzanie elementami linii** - elementy linii są zależne bezpośrednio od elementów sąsiadujacych tzn.: linia 1 oddziałuje na zwrotnicę 1, a zwrotnica 1 na linię 2. Jednak elementy te powinny przekazywać między sobą jedynie najistotniejsze informacje, kiedy powinny zacząć pracę czy jaki rodzaj trasy powinny realizować. Elementy powinny móc współpracować w sposób pozwalający na zrealizowanie (w przypadku tego konkretnego projektu) 3 rodzajów tras: o krótkiej, średniej bądź długiej długości.
- **Informacja zwrotna** - program powinien również dawać informację zwrotną np.: w postaci liczby zliczonych paczek czy obecnie wybranych tras dla konkretnych rozmiarów paczek

Powyższe założenia pozwalają na rozszerzenie projektu o dodatkowe linie, rozmiary paczek etc.

1.2 Działanie elementów

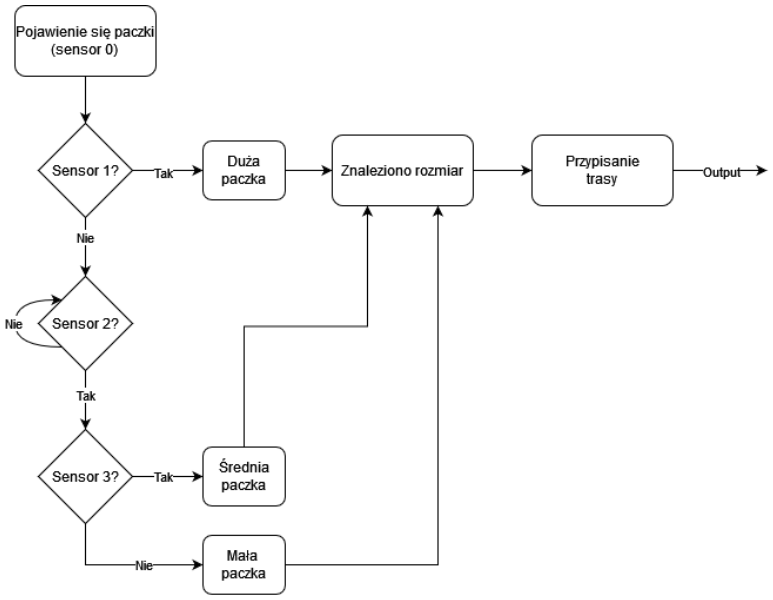
Do pracy algorytmu potrzebne jest kilka składowych elementów:

- Blok zwrotnicy
- Blok taśmy
- Blok kolejki
- Blok detekcji

W pierwszej kolejności należy więc zrozumieć jak powinny one działać

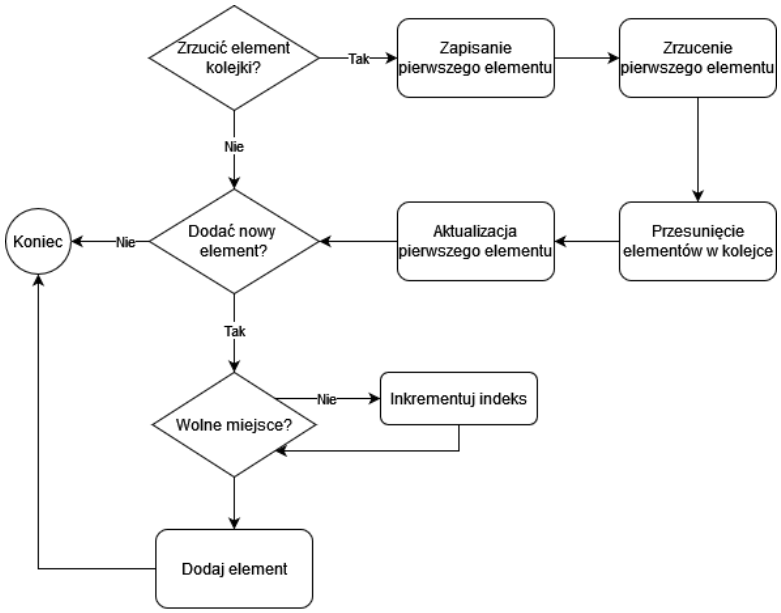
1.2.1 Blok detekcji

Oparty jest o sygnały z 4 sensorów znajdujących się na linii. Metodę detekcji paczek na podstawie ich sygnałów opracowano metodą prób i błędów.



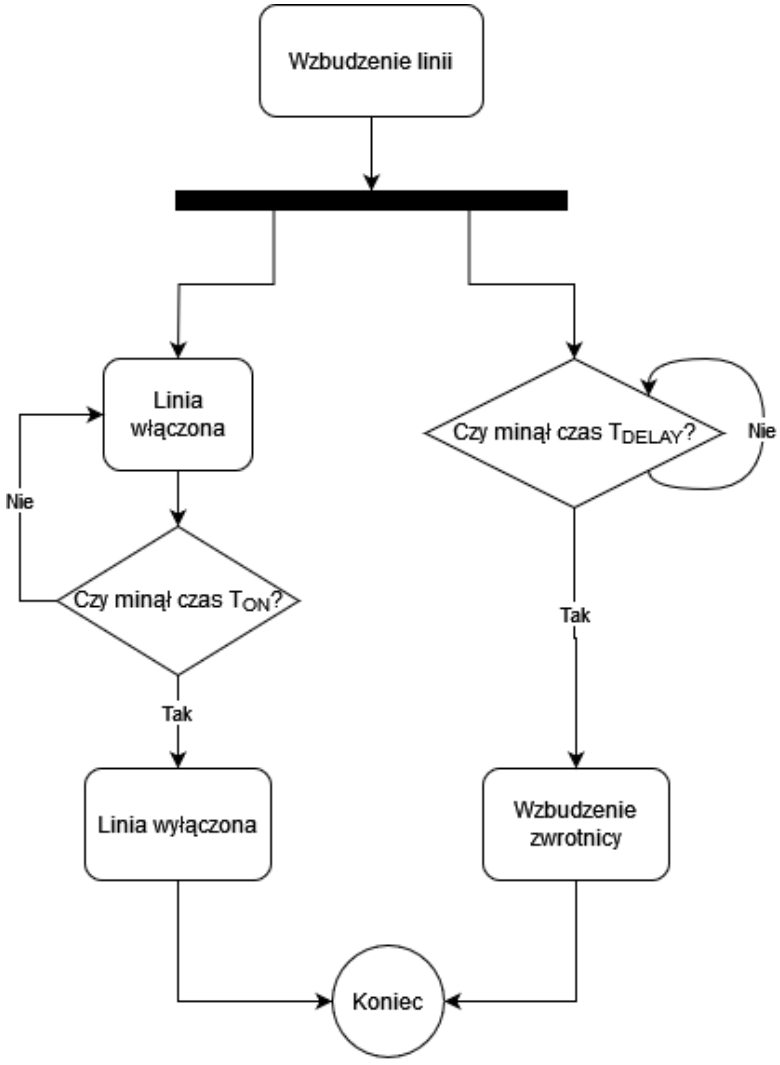
1.2.2 Blok kolejki

Kolejka powinna realizować dwie główne funkcje - zrzucania oraz dodawania elementów. Zrzucenie elementu polega na usunięciu pierwszego elementu z kolejki, a następnie przesunięciu całej reszty elementów o jeden 'do góry'. Dodawanie elementu natomiast powinno umieścić nowy obiekt pod pierwszym wolnym indeksem od góry.



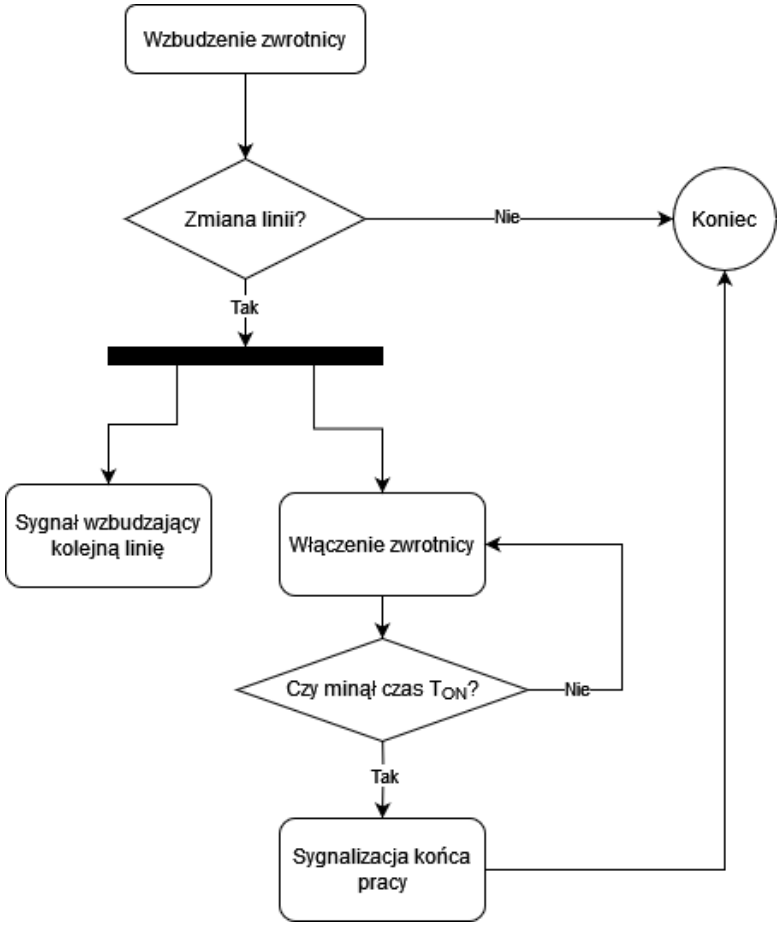
1.2.3 Blok linii

Linia po jej wzbudzeniu (każdorazowo) powinna działać przez czas T_{ON}, dodatkowo powinna generować w trakcie swojej pracy sygnały do wzbudzenia zwrotnic. Powinny one mieć miejsce po pewnym czasie T_{DELAY}.



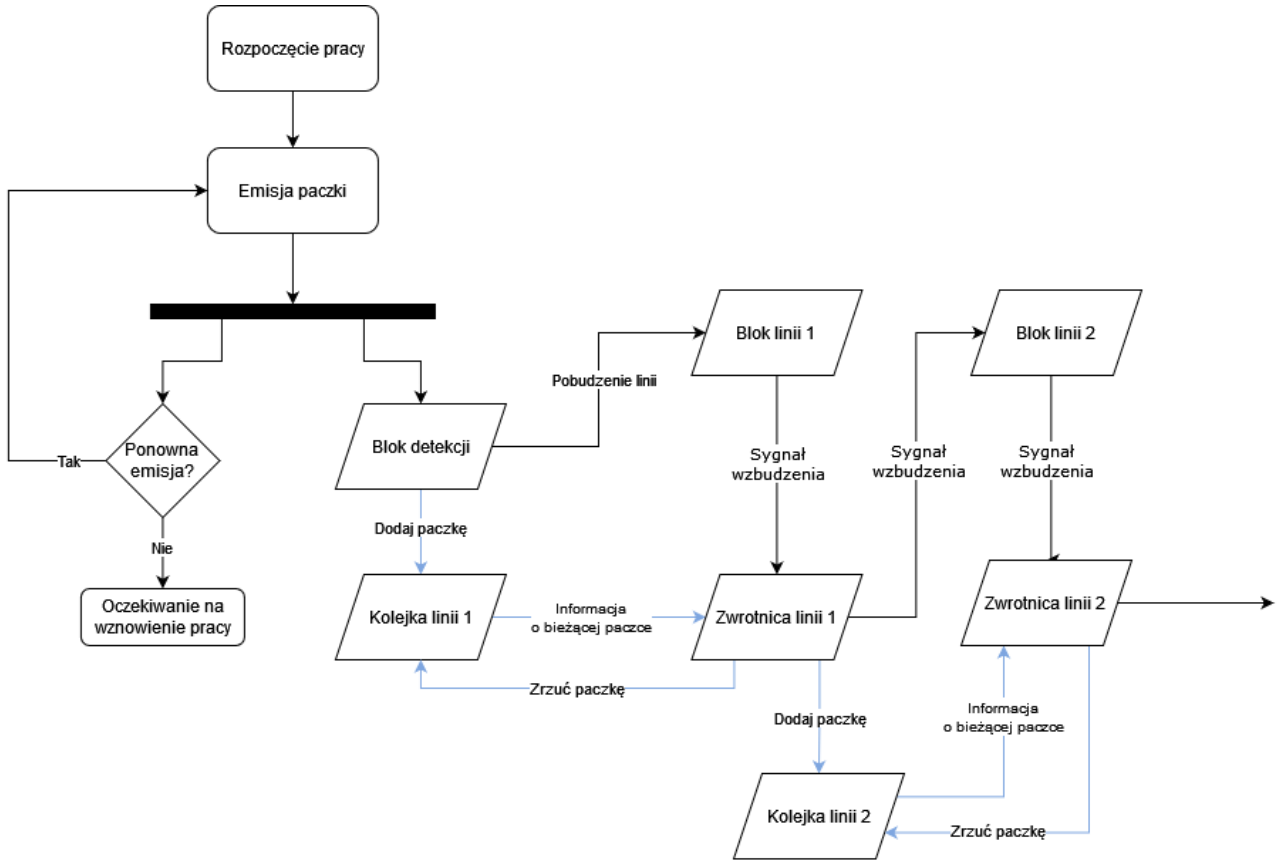
1.2.4 Blok zwrotnicy

Zwrotnica powinna przed swoim włączeniem zadecydować czy powinna przekierować paczkę. Jeśli tak, powinna włączyć samą siebie, ale również wzbudzić taśmę, na którą po czasie T_{ON} wjedzie paczka. Na samym końcu zwrotnica powinna zasygnalizować, że obsłużyła paczkę.



1.3 Całościowy schemat

Poniżej schemat bardziej ogólny z użyciem wcześniej wspomnianych elementów (oznaczone przekrzywionymi równoległobokami). Dodatkowo na niebiesko oznaczono sygnały związane wyłącznie z przekazywaniem informacji o paczkach.



Na rysunku widoczna jest aplikacja dla dwóch linii, jednak realizacja dla większej ilości byłaby analogiczna.

Część 2

Realizacja

2.1 Budowa programu

Idee stojące za projektem:

- **Skalowalność** - linia ma składać się z obiektów, które w łatwy sposób można powielić oraz które napisane są w sposób jak najbardziej ogólny i prosty do modyfikacji.
- **Ograniczona zależność** - elementy linii powinny być zależne bezpośrednio jedynie od obiektów sąsiadujących, np.: praca linii powinna być zależna od zwrotnicy bezpośrednio przed nią, ale nie od sygnałów z linii poprzedzającej. Pozwala to na dostosowanie pracy konkretnego odcinka trasy oraz odizolowanie problemów w programie.

Jako że działanie każdej linii jest podobne, ich obsługa jest dosyć powtarzalną czynnością. Zdecydowano o użyciu bloków funkcyjnych, by program był przejrzysty i jak najkrótszy. Zastosowanie bloków funkcyjnych zapewnia także swobodę dobrania wygodniejszego języka: tekstowego lub graficznego, w zależności od potrzeb.

2.2 Zastosowane bloki funkcyjne

2.2.1 Obsługa paczek

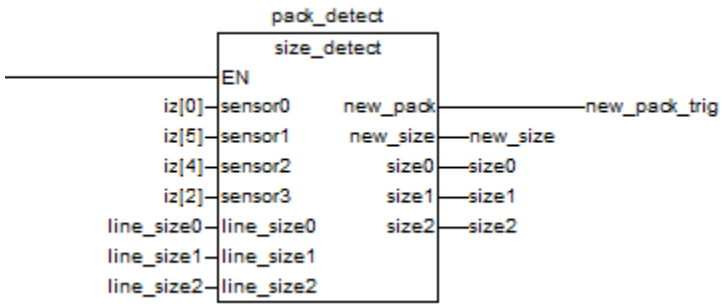
Wykrywanie paczek Za kategoryzowanie paczek odpowiada blok funkcyjny ***size_detect***. Na podstawie zaobserwowanych aktywowanych czujników - zdalnych wejść, klasyfikuje paczkę jako małą, średnią lub też dużą. Następnie zgodnie z żądaniem użytkownika przypisuje wartość odpowiadającą wariantowi trasy, którą ma pojechać paczka. Po klasyfikacji paczki wysyła sygnał *new_pack*.

Opis wejść i wyjść bloku *size_detect*

Nazwa	Rodzaj	Opis
sensor0-3	Input - BOOL	Wejścia na sygnały sensorów pozwalających wykryć rozmiar paczki
line_size0-2	Input - WORD	Wejścia mówiące jaką trajektorią mają podążać paczki, w kolejności duża, średnia, mała.
new_pack	Output - BOOL	Sygnał w postaci impulsów informujący o wykryciu nowej paczki
new_size	Output - WORD	Wartość liczbowa $new\_size \in < 0,\ 2 >$ definiująca rozmiar nowej paczki
size0-2	Output - BOOL	Linie przyjmują stan wysoki, kiedy wykryto paczkę o korespondującym rozmiarze. Wyjście pomocnicze do zliczania ilości wykrytych paczek konkretnego typu.

```
size0_trg(CLK := sensor0);
size2_trg(CLK := sensor2);
reset_trg(CLK := sensor3);

IF size0_trg.Q AND sensor1 THEN
        new_size := line_size0;
        size0  := TRUE;
        size1 := FALSE;
        size2 := FALSE;
        size0_det := TRUE;

        new_pack := TRUE;
ELSIF size2_trg.Q AND NOT size0_det  THEN
        IF sensor3 THEN
                new_size := line_size1;
                size1  := TRUE;
                size0 := FALSE;
                size2 := FALSE;
                size0_det := FALSE;
```

```
                new_pack := TRUE;
        ELSE
                new_size := line_size2;
                size2  := TRUE;
                size1 := FALSE;
                size0 := FALSE;
                size0_det := FALSE;

                new_pack := TRUE;
        END_IF;
ELSIF reset_trg.Q THEN
        size0_det := FALSE;
        size1 := FALSE;
        size1 := FALSE;
        size0 := FALSE;
ELSE
        new_pack := FALSE;
        size1 := FALSE;
        size1 := FALSE;
        size0 := FALSE;
END_IF;
```

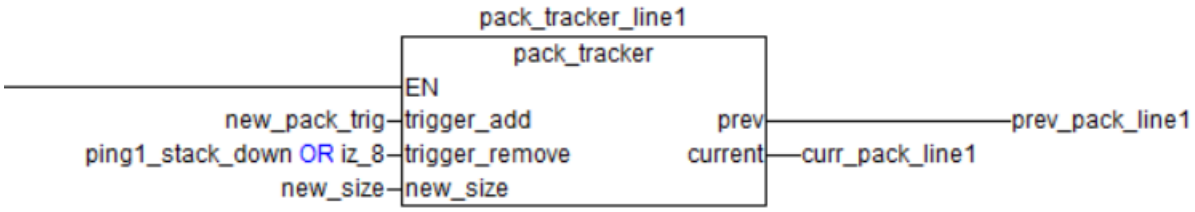


Wygląd bloku użytego w programie

Śledzenie paczki na linii Każda taśma, której zwrotnica potrzebuje informacji na temat rozmiaru obecnie przetwarzanej paczki jest 'wyposażona' w blok ***pack_tracker***. Pełni on funkcję kolejki (FIFO), domyślnie może przechowywać informację o 5 elementach jednak w łatwy sposób może być to zmodyfikowane. Jako wartość równoważną do braku paczki przyjęto 255. Zatem jeżeli linia jest pusta, wszystkie miejsca w kolejce oznaczone są 255, w przypadku wejścia nowej paczki otrzymuje ona pierwsze miejsce w kolejce, a reszta miejsc jest 'wyzerowana' 255 oczekując na następne pakunki.

Opis wejść i wyjść bloku *pack_tracker*

Nazwa	Rodzaj	Opis
trigger_add	Input - BOOL	Sygnał, którego rosnąca krawędź powoduje dodanie paczki rozmiaru *new_size* do kolejki
trigger_remove	Input - BOOL	Sygnał, którego rosnąca krawędź powoduje zrzucenie z kolejki pierwszy element
new_size	Input - WORD	Wartość liczbowa $new\_size \in < 0, 2 >$ definiująca rozmiar nowej paczki
current	Output - WORD	Przyjmuje wartość pierwszego elementu kolejki
prev	Output - WORD	Przyjmuje wartość ostatnio zrzuconego elementu kolejki



Kod w ST bloku

```
stack_up_trig(CLK := trigger_add);
stack_down_trig(CLK := trigger_remove);

IF stack_init THEN
        FOR i :=1 TO stack_size DO
                stack[i] := 255;
        END_FOR;
        stack_init := FALSE;
END_IF;

IF stack_down_trig.Q THEN
        prev := stack[1];
```

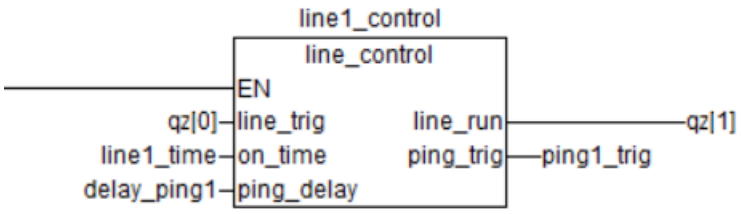
```
        FOR i := 1 TO stack_size-1 DO
                IF stack[i] = 255 THEN
                        EXIT;
                END_IF;

                stack[i] := stack[i+1];
        END_FOR;
        current := stack[1];
END_IF;

IF stack_up_trig.Q THEN
        FOR i := 1         TO stack_size-1 DO
                IF stack[i] = 255 THEN
                        stack[i] := new_size;
                        IF i = 1 THEN
                                current := new_size;
                        END_IF;

                        EXIT;
                END_IF;
        END_FOR;
END_IF;
```

2.2.2 Obsługa mechanizmów linii

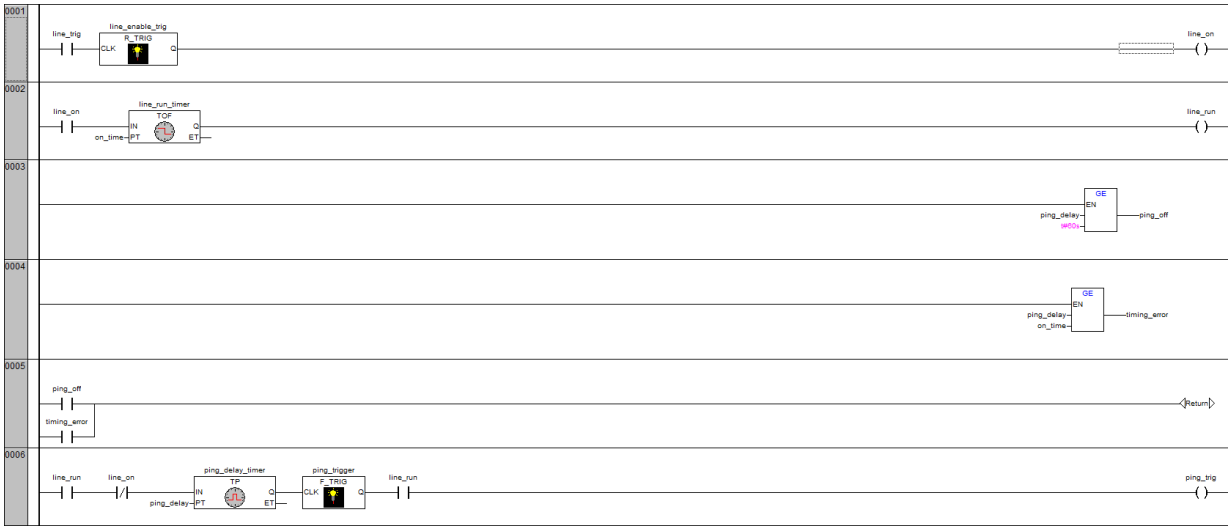
Obsługa taśmy Po zadaniu sygnału *line trig*, linia dostaje sygnał triggera, który uruchamia timer TOF i w ten sposób 'pobudza' zmienną *line_run* na żądaną minimalną ilość czasu. W wypadku pojawienia się na linii paczki tego samego rodzaju, sygnał zostanie podtrzymany.
Dodatkowo linia przekazuje sygnał wzbudzający do zwrotnicy, który ma miejsce po czasie *ping_delay* od chwilii *line trig*. By blok funkcjonował poprawnie musi być spełniony warunek $ping\_delay < line\_on$ oraz $ping\_delay < 60$

Dokładniejszy opis wejść i wyjść bloku *line_control*

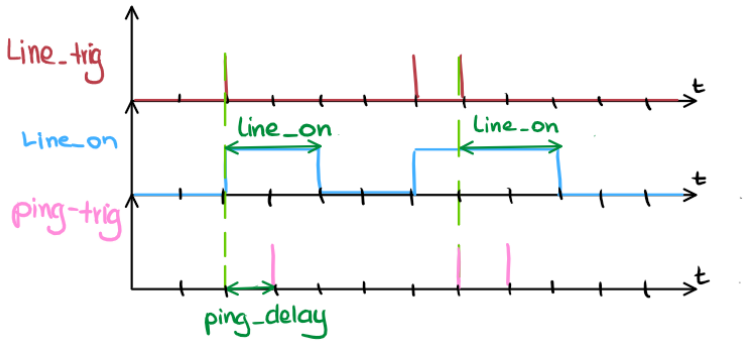
Nazwa	Rodzaj	Opis
line_trig	Input - BOOL	Sygnał, którego narastające zbocze aktywuje linię
on_time	Input - WORD	Wartość czasowa wyrażająca jak długo ma pracować linia po ostatnim załączeniu
ping_delay	Input - WORD	Wartość czasowa wyrażająca z jakim opóźnieniem (względem załączenia linii) ma być aktywowana zwrotnica
line_run	Output - BOOL	Jest w stanie wysokim, gdy linia jest załączona
line_trig	Output - BOOL	Sygnał w formie impulsowej, który pozwala na załączenie zwrotnicy po czasie *ping_delay* od sygnału *line_trig*



Wygląd bloku użytego w programie



Wnętrze bloku funkcyjnego w LD



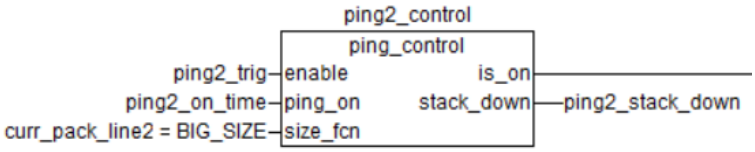
Rys. 2.1: Działanie zobrazowane ogólnymi przebiegami czasowymi

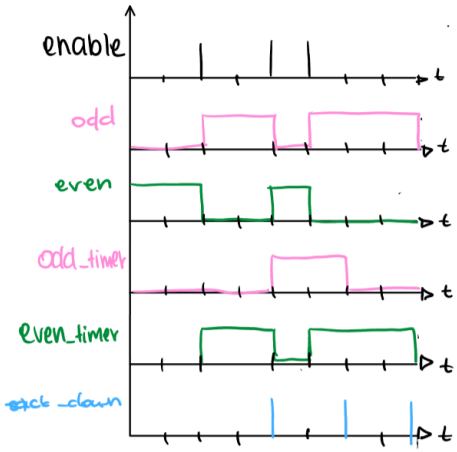
Obsługa zwrotnicy Podobnie jak przy obsłudze taśmy zwrotnica jest uruchamiana za pomocą sygnału *enable* i jest włączana na czas minimalny *ping_on* z możliwością podtrzymania pracy. Istotną różnicą jest jednak obecność wejścia *size_fcn*, odpowiada ono za dostarczenie infomracji o tym czy paczka powinna być przekierowana (stan wysoki) czy nie (stan niski). W przypadku stanu niskiego pomimo obecności sygnału *enable* zwrotnica się nie uruchomi.

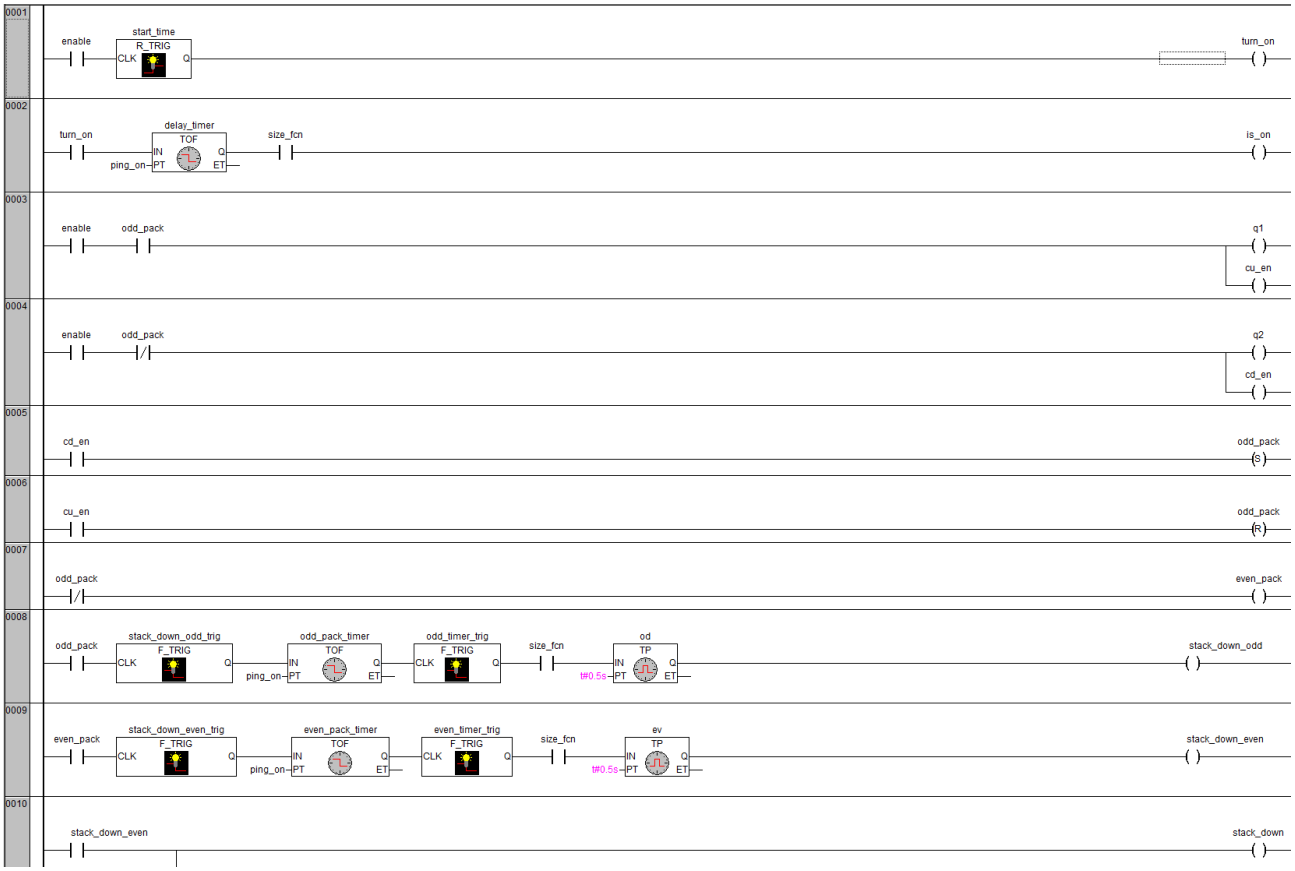
Komentarza wymaga zastosowanie naprzemiennie setowanych i resetowanych sygnałów. By system działał poprawnie zwrotnica powinna każdorazowo informować o zakończeniu 'przerabiania' paczki, okazało się to problematyczne przy ciągłej pracy zwrotnicy. Rozwiązaniem jest obsługa 'naprzemienna' paczek, można to sobie wyobrazić przypisując kolejnym paczkom naprzemiennie wartości 0 oraz 1. Paczki 0 oraz 1 są obsługiwane oddzielnie dzięki czemu uniknięto przypadkowych podtrzymań sygnału *stack_down* jakby to miało miejsce przy zastosowaniu wyłącznie jednego timera TOF.

Dokładniejszy opis wejść i wyjść bloku *ping_control*

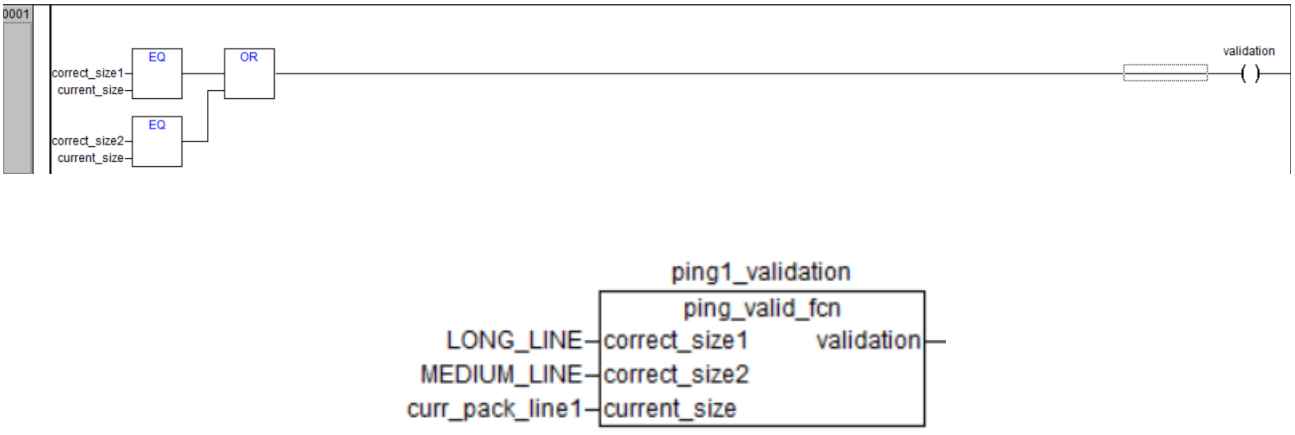
Nazwa	Rodzaj	Opis
enable	Input - BOOL	Sygnał, którego narastające zbocze aktywuje zwrotnicę
ping_ong	Input - WORD	Wartość czasowa wyrażająca jak długo zajmuje zwrotnicy przeniesienie paczki na sąsiednią linię
size_fcn	Input - BOOL	Sygnał informujący czy obecnie obsługiwana paczka powinna zostać przekierowana (stan wysoki)
is_on	Output - BOOL	Jest w stanie wysokim, gdy zwrotnica jest załączona
stack_down	Output - BOOL	Sygnał w formie impulsowej, który informuje o 'przetworzeniu' obecnej paczki





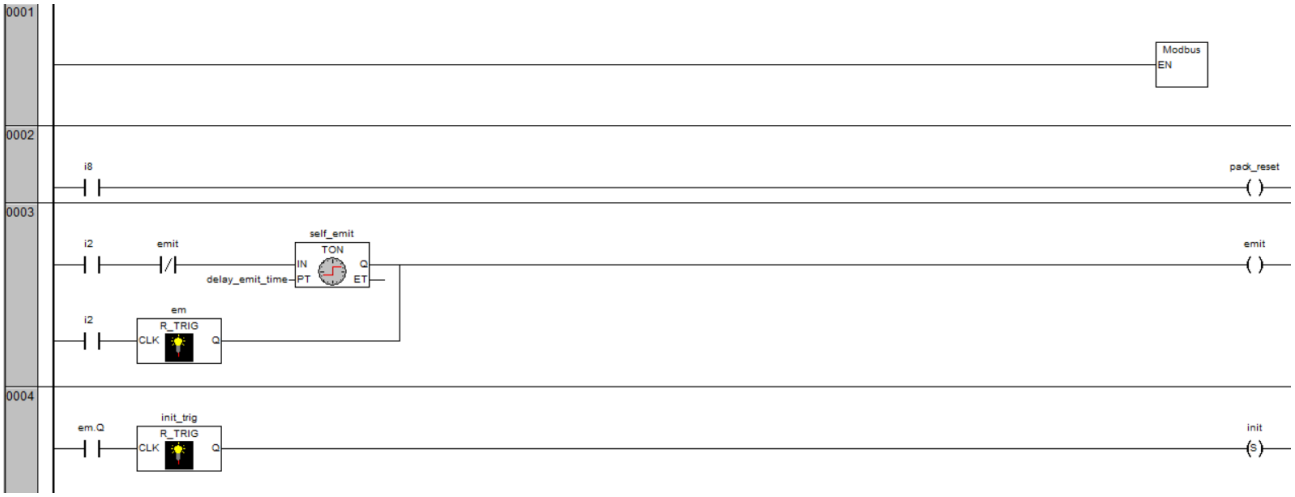


Zarządzanie zwrotnicą Rolę decyzyjną w procesie załączania zwrotnicy pełni funkcja pomocnicza *ping_valid_fcn*. Swoje konkretne zastosowanie znajduje w obsłudze pierwszej zwrotnicy, gdzie ma za zadanie upewnić się, że jedynie paczki mające jechać dalej, przejadą. Jednak, gdyby zainstniała taka potrzeba funkcję można zastosować w dowolnej zwrotnicy.

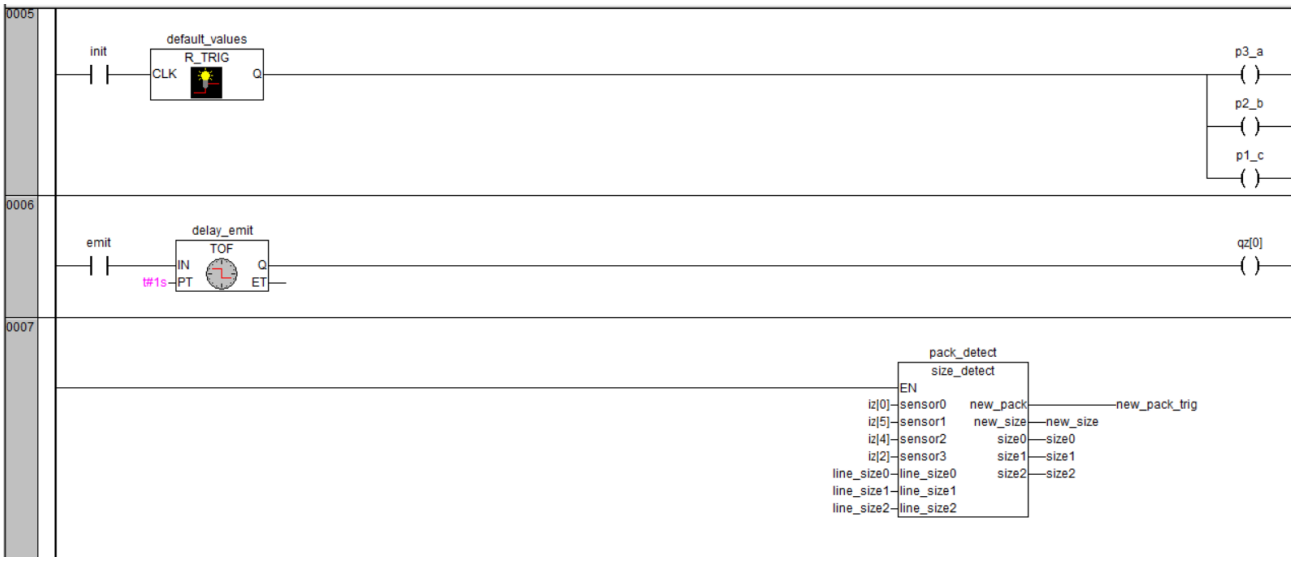


Użycie funkcji w programie do obsługi pierwszej zwrotnicy

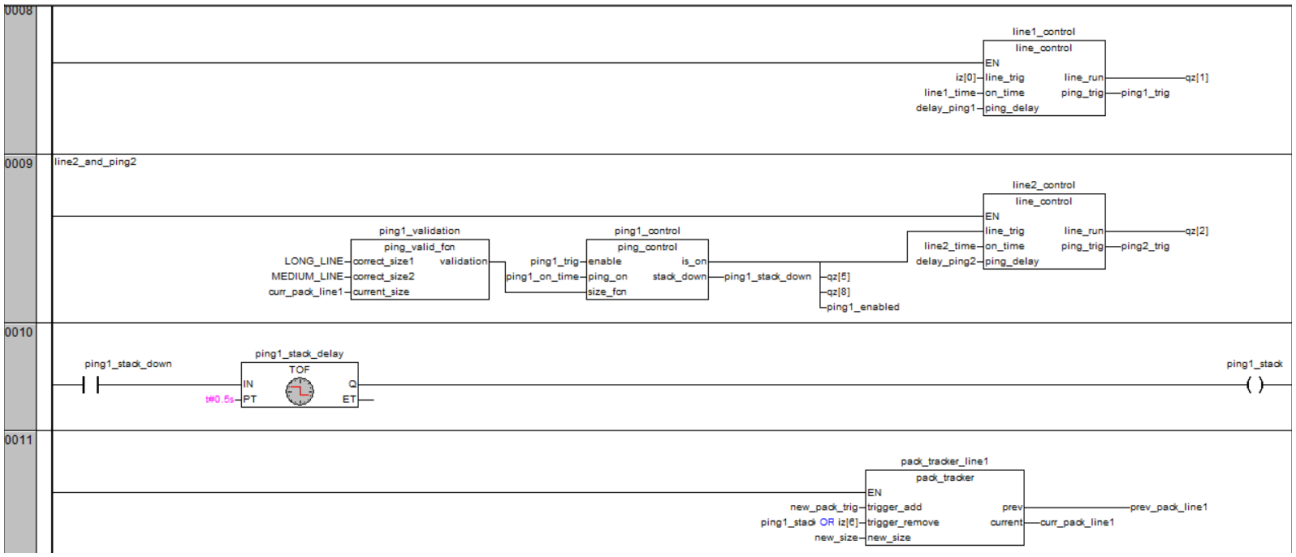
2.3 Pełny program



Jak widać, początek programu nie należy do najbardziej skomplikowanych. Przycisk wejście i2 rozpoczyna cykliczne emitowanie paczek zgodnie z początkowym fragmentem opracowanego algorytmu. Dodano także opcję *pack_reset*, która resetuje liczniki paczek.



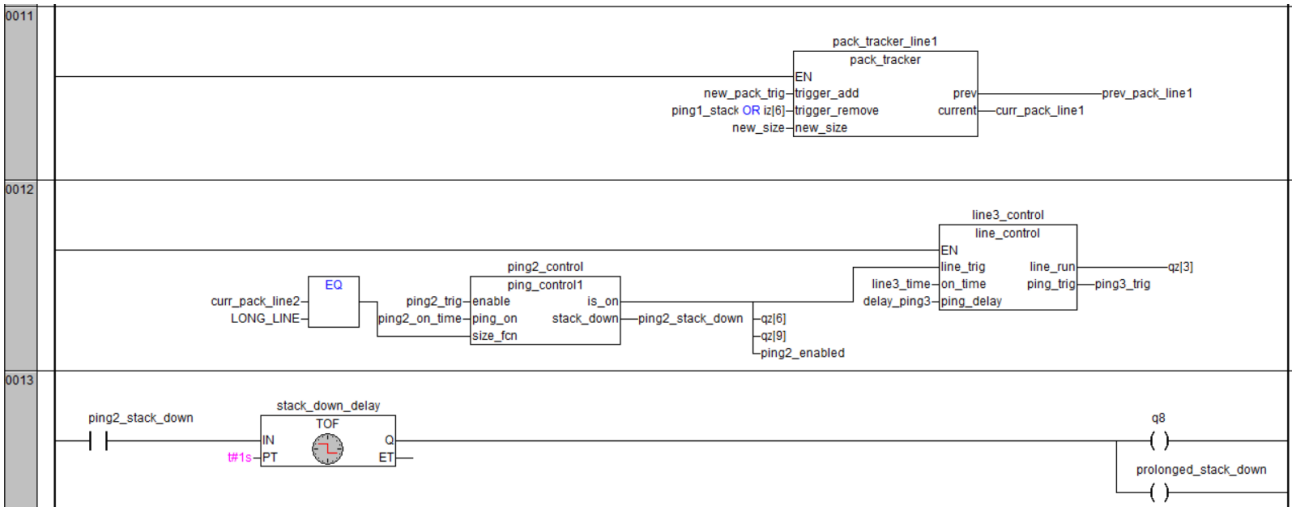
Następnie następuje zainicjowanie tras dla wszystkich rozmiarów, a także uruchomienie pierwszej linii. Widoczna jest realizacja detekcji paczek za pomocą bloku *size_detect*. Szczególną uwagę należy zwrćcić na sygnał *new_pack_trig*, który w dalszej części programu odpowiada za dodawanie paczek kolejki linii 1.



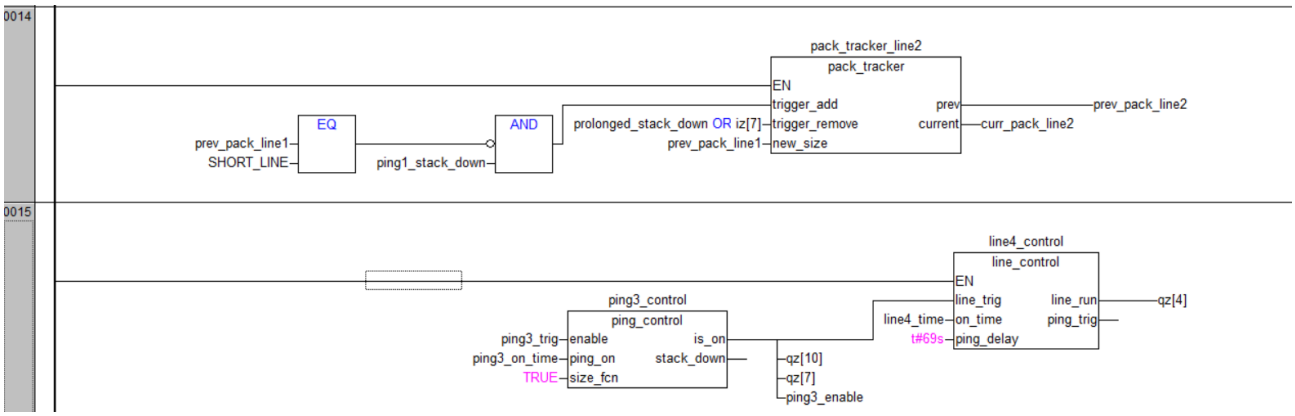
Jak widać, dzięki zastosowanym blokom funkcyjnym dodawanie kolejnych linii oraz ich konfiguracja są bardzo proste. Linia pierwsza jako potrzebna każdej trasie uruchamiana jest tak szybko, jak tylko czujnik iz[0] wykryje na niej obecność jakiegokolwiek rozmiaru paczki. Następnie sprytnie połączone bloki *ping1_validation* oraz *ping1_control* włączają zarówno zwrotnicę, ale także razem z nią kolejną linię - oczywiście jedynie, jeżeli trasa nie kończy się na pierwszym pojemniku.

Widoczny jest również blok *pack_tracker* obsługujący kolejkę linii 1. Zarządzany jest przez sygnał z *size_detectora* oraz sygnał mówiący o zakończeniu pracy przez zwrotnicę 1.

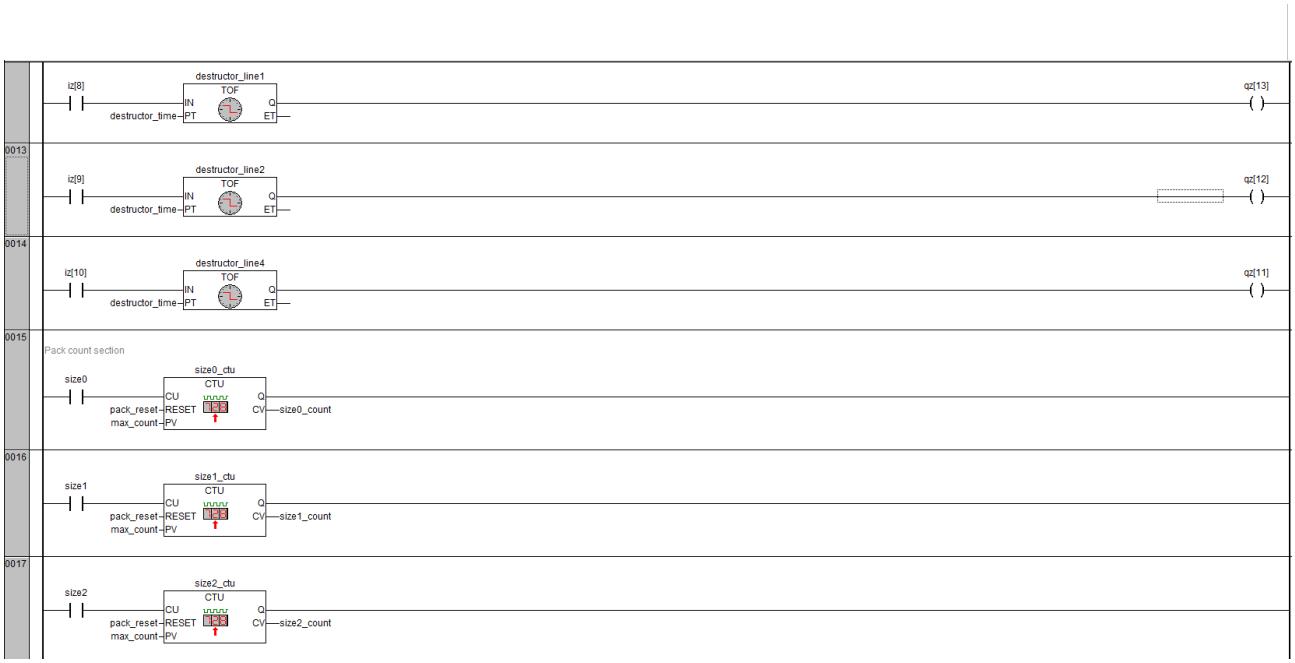
Warto zauważyć, że przedłużono sygnał *ping1_stack_down*. Sygnały zostały przedłużone blokiem TOF, by można było łatwiej je zaobserwować podczas działania programu, taki sam zabieg został zastosowany w obsłudze drugiej 'decyzyjnej' zwrotnicy.



Analogiczny mechanizm kontroli zwrotnicy i drugiej linii z przedłużonym sygnałem. Jak wyżej, dodane przedłużenie sygnału *stack_down* z dodatkową informacją w postaci wyjścia *q8* do obserwacji tego sygnału.

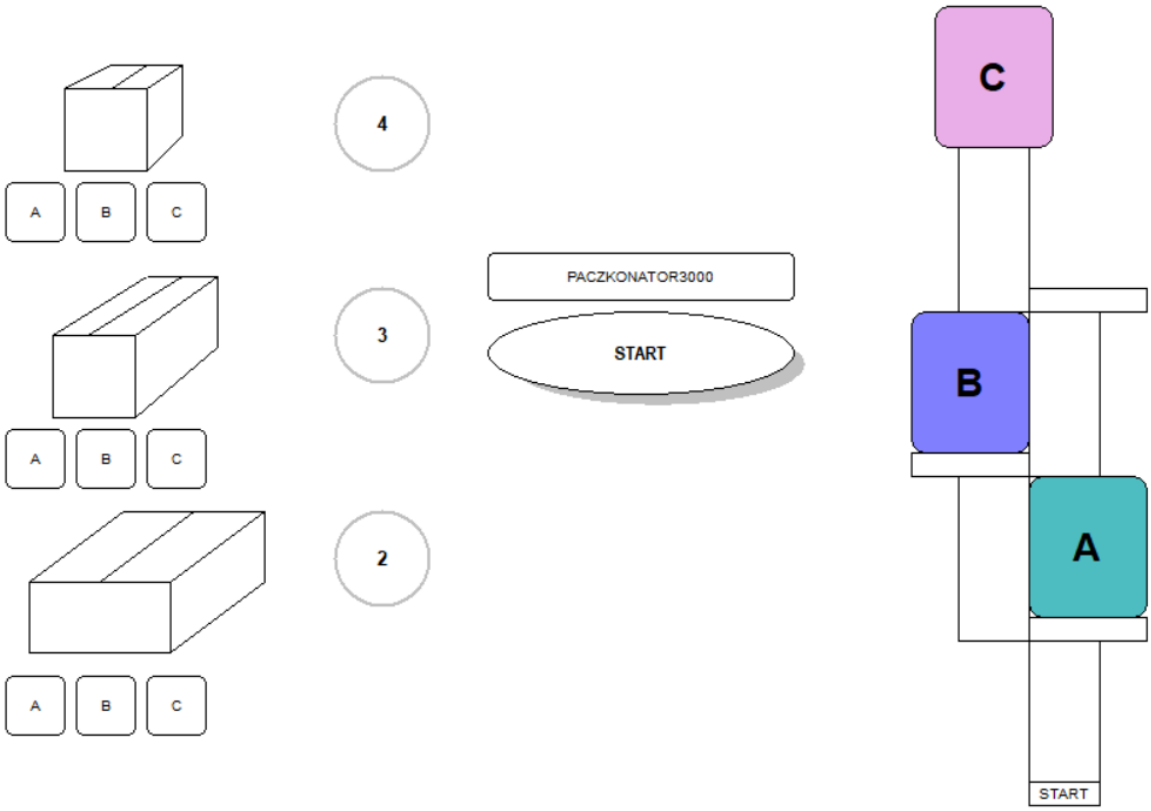


W końcu, obsługa kolejki drugiej zwrotnicy, trzeciej 'niedecyzyjnej' zwrotnicy oraz taśmy za nią. Tym sposobem obsłużone zostały wszystkie linie w zależności od żądanego rozmiaru paczek.



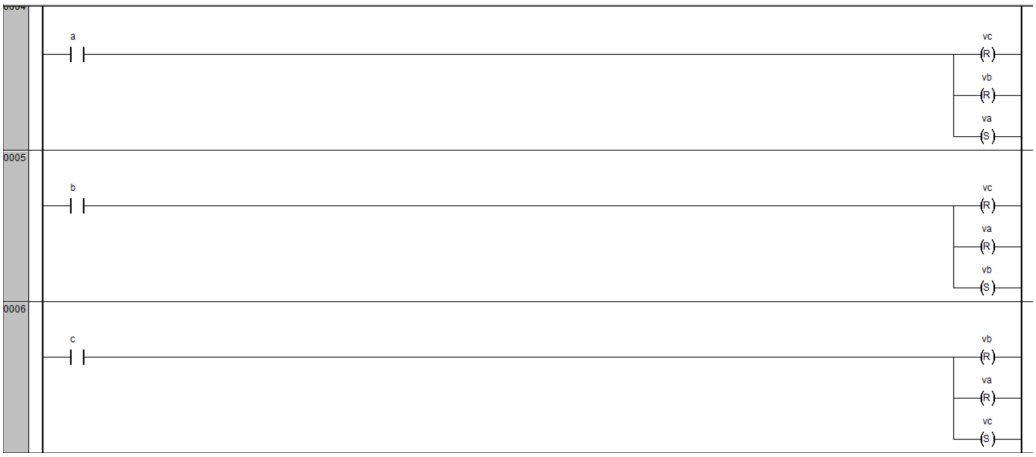
W końcowej części programu umieszczono obsługę niszczarek oraz pomocnicze countery wskazujące ilość wyemitowanych paczek każdego typu. Po nich znajdują się już jedynie bloki przeznaczone do obsługi przycisków GUI, opisane w następnej sekcji.

2.3.1 Wizualizacja

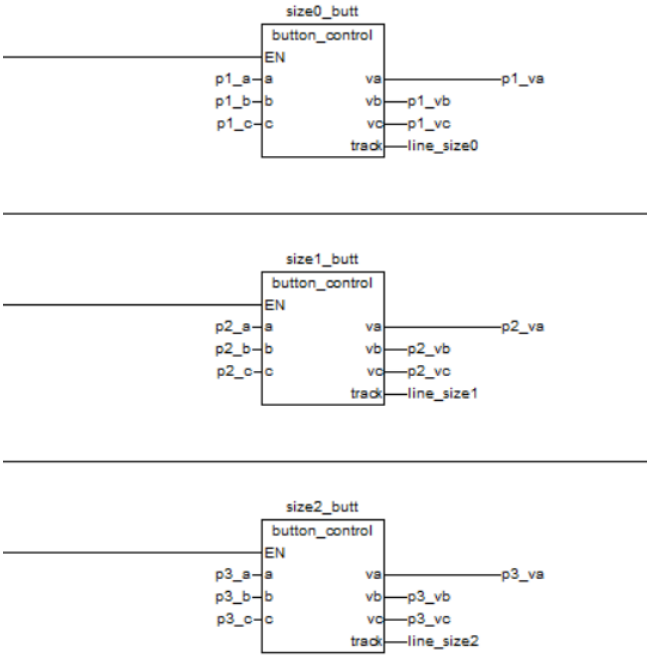


Widok GUI

Do obsługi programu przydaje się panel wizualizacyjny. Na schemacie zaznaczone są nazwy nadane anihilatorom oraz intuicyjne zaznaczanie dogodnego celu paczki pod schematycznym oznaczeniem rozmiaru. Obok symboli znajdują się również liczniki nadanych paczek.
Aby obsługa za pomocą wirtualnych przycisków była możliwa, należało połączyć je ze zmiennymi w programie. Jak większość operacji w zastosowanym podejściu, było to zadanie powtarzalne dla każdego rozmiaru, dlatego tę kwestię również rozwiązano zastosowaniem bloku funkcyjnego.



Prosta budowa przyporządkowania przycisków do tras oraz następnie tychże do rozmiarów paczek.



Rys. 2.2: Zastosowanie bloków w programie. Każde wejście odpowiada jednemu z dziewięciu przycisków.