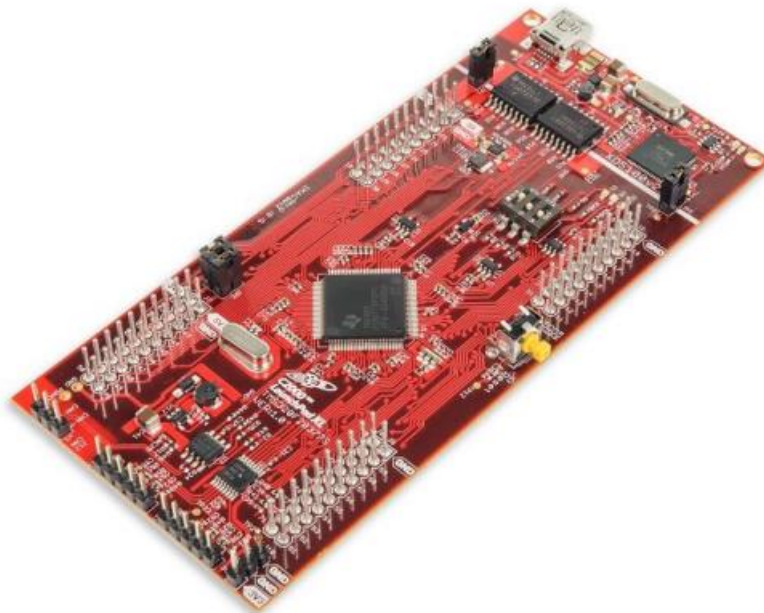
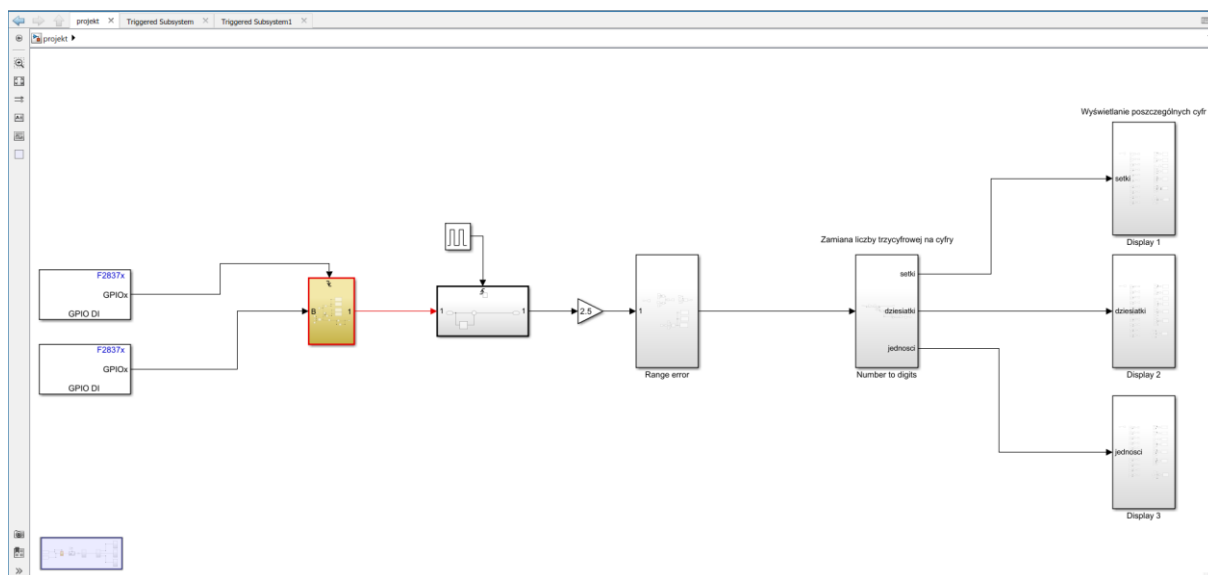


1. Polecenie

Stworzenie programu do mikrokontrolera przedstawionego poniżej, który wyświetla prędkość obrotową (za okres 1s) impulsatora wyrażoną w obr/min oraz wskazuje kierunek wirowania przy wykorzystaniu diod LED. Przyjmujemy, że uzyskuje się 24 impulsy na jeden obrót.



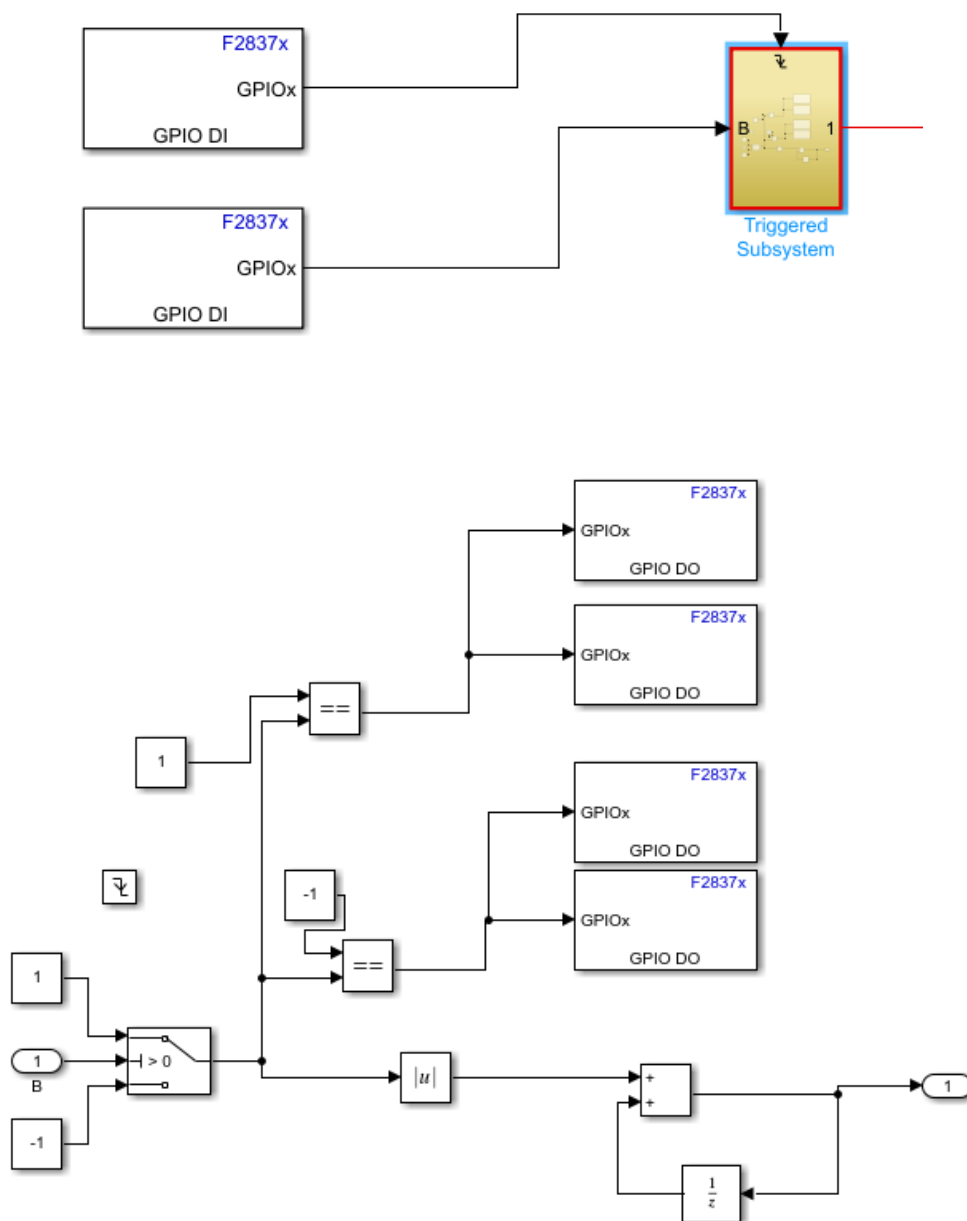
2. Wykonanie projektu w programie Matlab-Simulink



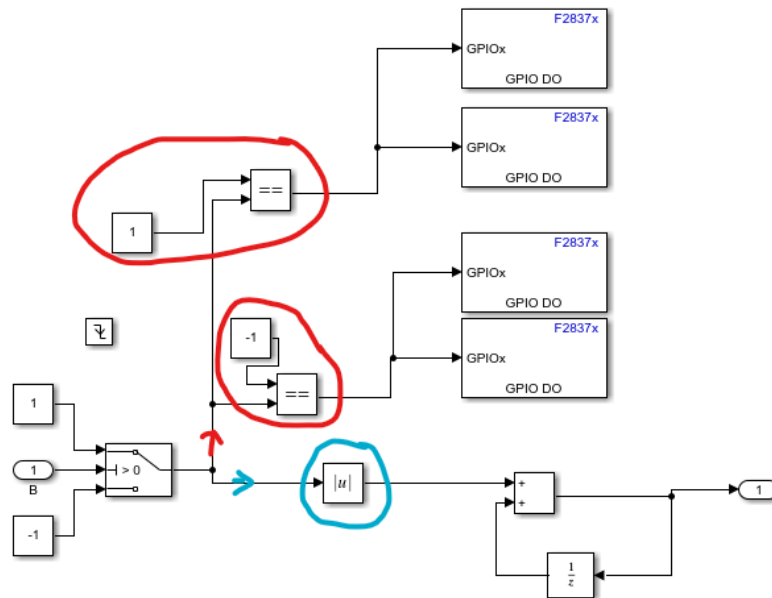
Rysunek 1 Schemat projektu w programie Matlab-Simulink

3. Zasada działania

Na podstawie sygnałów GPIO130 oraz GPIO131 odpowiedzialnych za działanie impulsatora otrzymujemy sygnały, które w zależności od kierunku obrotu zadają wartość 1 lub -1. Odpowiada za to ten fragment schematu blokowego:



W subsystemie, przy pomocy bloczka *Switch* otrzymuje się odpowiednio 1 lub -1 w zależności od kierunku obrotu. Następnie sygnał trafia na odpowiednio na:

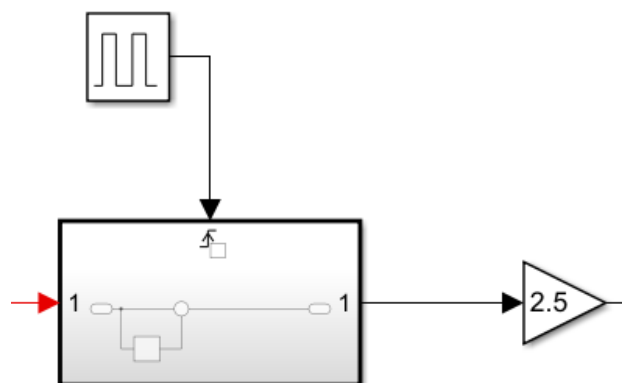


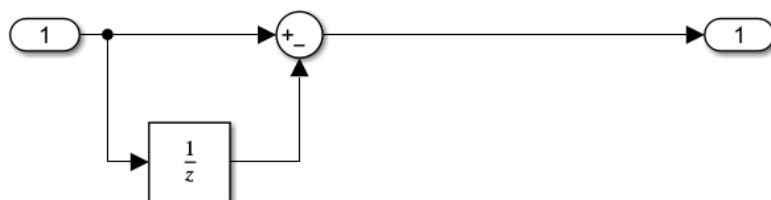
Bloczki *Relational Operator* (oznaczone kolorem czerwonym), które porównują wartość impulsu z 1 , gdy impulsator obraca się w prawo oraz -1, gdy impulsator obraca się w lewo. Bloczki *Relational Operator* ustawiają stan wysoki na wyjściach w postaci dwóch diod LED GPIO14, GPIO15 oraz GPIO10, GPIO11 w zależności od kierunku obrotu impulsatora.

Sygnal trafia również na bloczek *Abs* (oznaczone kolorem niebieskim), który odpowiada za wartość bezwzględną sygnału wchodzącego, czyli nie ważne od kierunku obrotu impulsatora zawsze będzie wyświetlana dodatnia wartość prędkości obrotowej na wyświetlaczu.

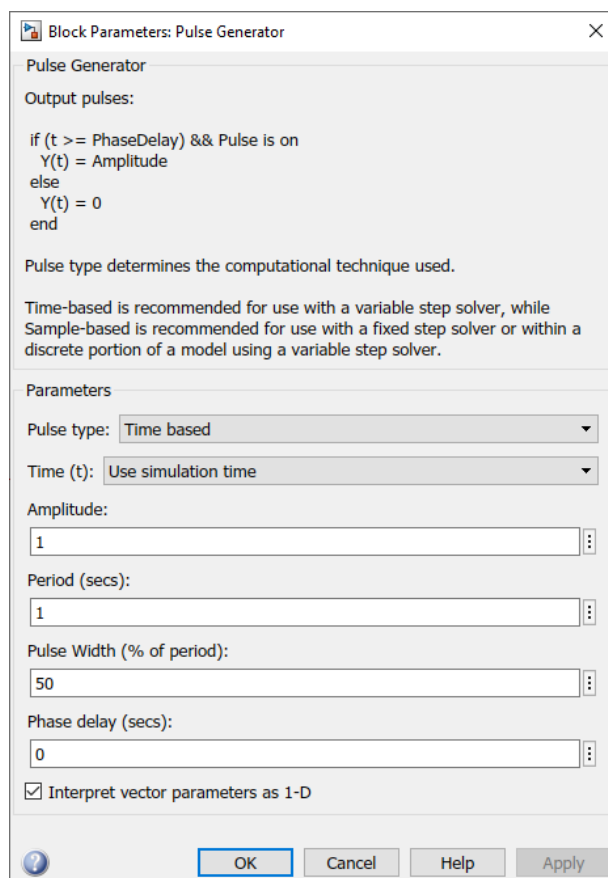
Następnie wartości są dodawane do siebie (bloczek *Add*) z poprzednią próbką przy pomocy (bloczek *Unit Delay*) i podawane na wyjście z subsystemu.

Następnie suma wartości zadanych trafia na kolejny triggerowany subsystem:

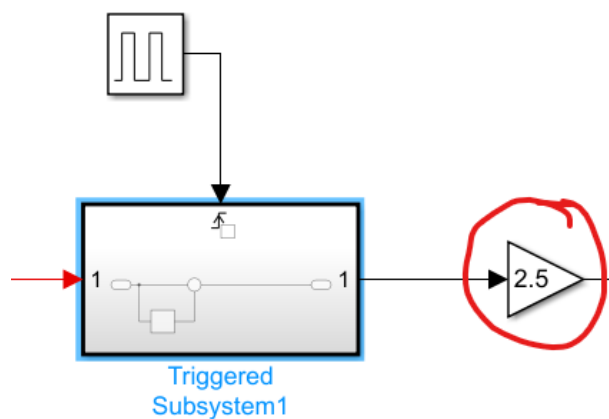




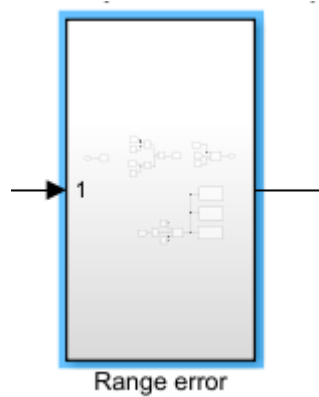
Subsystem ten jest triggerowany zboczem narastającym podawanym przy pomocy bloczka *Pulse Generator* tak aby wyświetlane wartości prędkości obrotowej na wyświetlaczu były odejmowane od poprzednich wartości w celu restartowania wskazań na wyświetlaczu przy braku obrotu impulsatora.



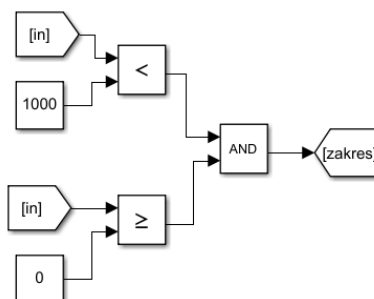
Generator impulsów jest ustawiony na restartowanie wskazań na wyświetlaczu o okresie 1s.



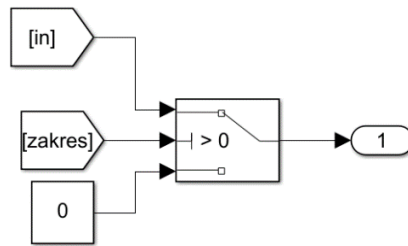
Wartości wychodzą z subsystemu do bloczka *Gain* który przemnaża wartość zadaną o 2,5 aby zamienić ilość impuls/obr impulsatora na obr/min.



Bloczek *Range error* określa zakres w jakim zawiera się liczba. Musi być on od 0 do 999 gdyż wyświetlacz 7 segmentowy z którego korzystaliśmy podczas zajęć umożliwia wyświetlanie liczb 3 cyfrowych (od 0 do 999).



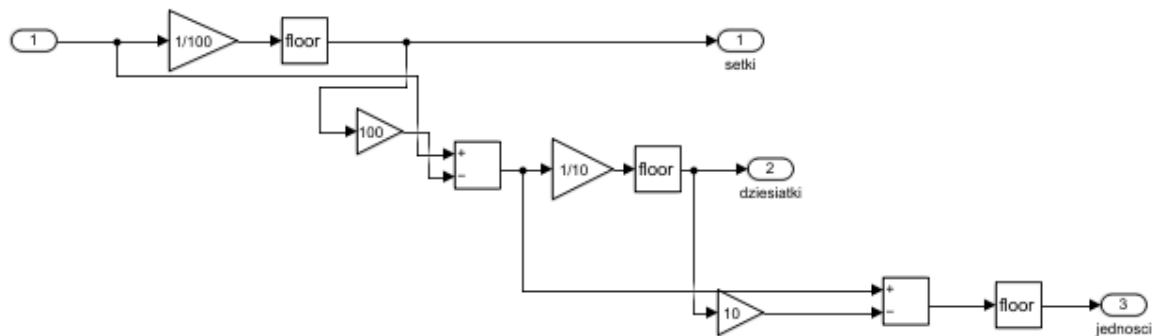
Ten subsystem sprawdza nam czy dany bloczek jest z zakresu 0-999



Ten subsystem daje na wyjściu wartość wejściową jeśli liczba należy do zakresu (0 999).



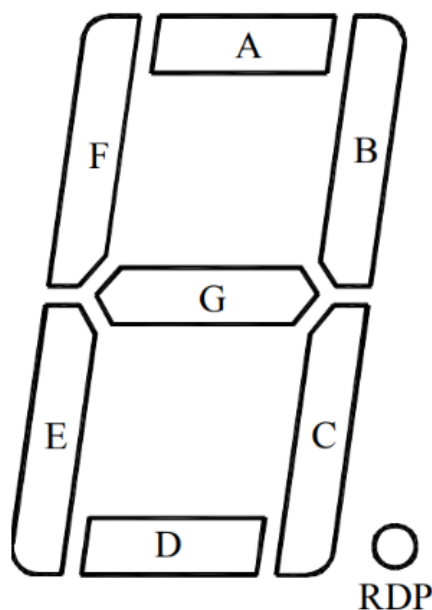
Bloczek *numer to digits* przekształca liczby na 3 liczby odpowiadające setek, dziesiątek i jedności



„Środek ” bloczku *numer to digits*



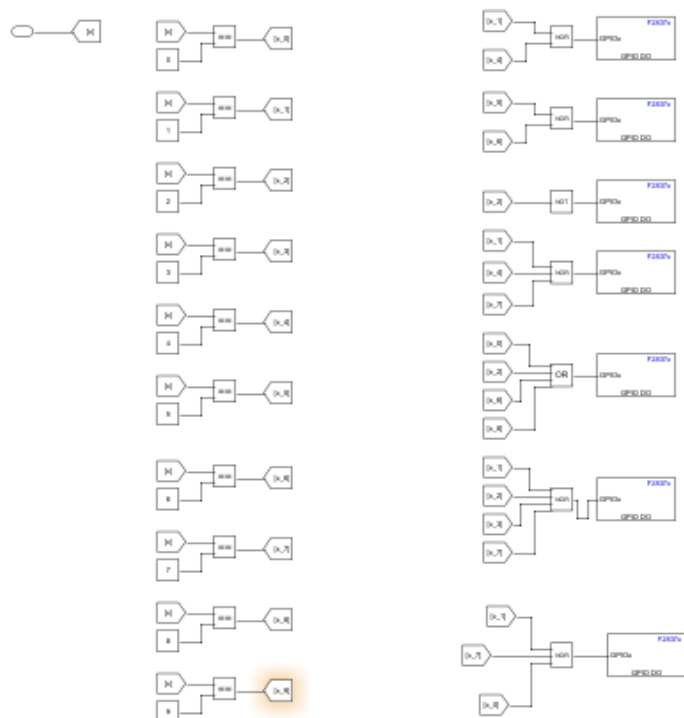
Te trzy bloczki bloczki reprezentują liczby setek, dziesiątek i jedności na wyświetlaczu segmentowym. Do każdej liczby określamy segment który powinien się świecić za pomocą bloczków OR, NOR, NOT.



Wyświetlacz siedmiosegmentowy

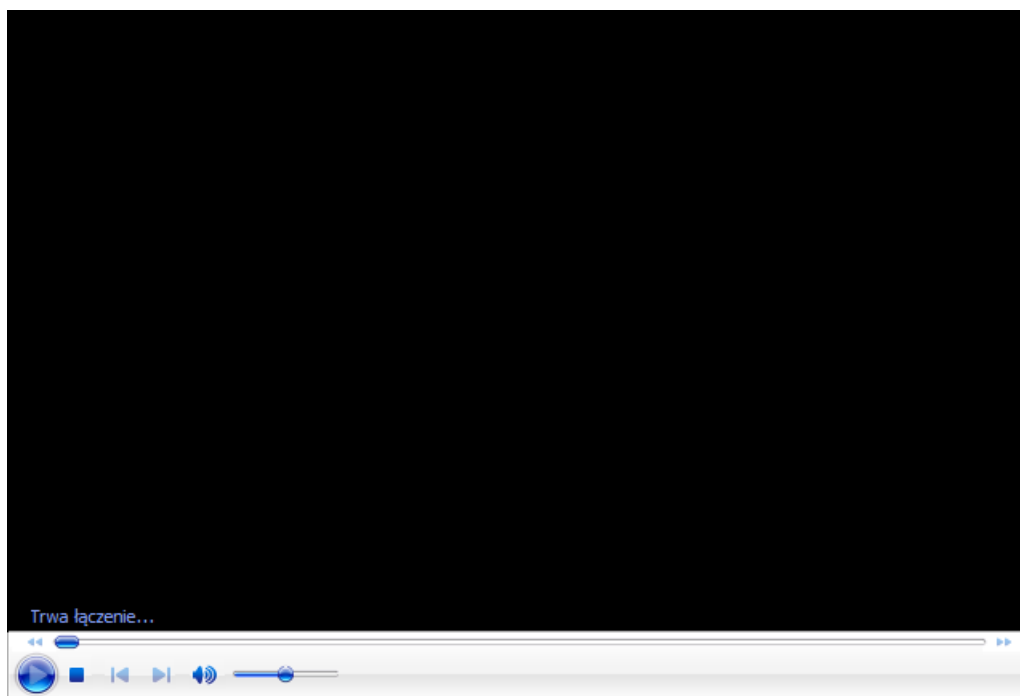
Tabela prawdy do sterowania 7 segmentowym wyświetlaczem

Segment	A	B	C	D	E	F	G	Liczba HEX
	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony		0
		Zapalony	Zapalony					1
	Zapalony	Zapalony	Zapalony			Zapalony	Zapalony	2
	Zapalony	Zapalony	Zapalony	Zapalony			Zapalony	3
		Zapalony	Zapalony			Zapalony	Zapalony	4
	Zapalony		Zapalony	Zapalony		Zapalony	Zapalony	5
	Zapalony		Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	6
	Zapalony	Zapalony	Zapalony					7
	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	8
	Zapalony	Zapalony	Zapalony	Zapalony		Zapalony	Zapalony	9
	Zapalony	Zapalony	Zapalony		Zapalony	Zapalony	Zapalony	A-10
	Zapalony	Zapalony		Zapalony	Zapalony	Zapalony	Zapalony	B-11
	Zapalony			Zapalony	Zapalony	Zapalony		C-12
	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony	Zapalony		D-13
	Zapalony			Zapalony	Zapalony	Zapalony	Zapalony	E-14
	Zapalony				Zapalony	Zapalony	Zapalony	F-15



Wnętrze subsystemu sterującego wyświetlaczem

Poniżej w pliku video zostało przedstawione działanie opisanego programu (żeby go włączyć należy kliknąć na niego dwa razy LPM).



LINK DO VIDEO W PRZYPADKU GDYBY FILM NIE DZIAŁAŁ

<https://we.tl/t-mThJ1Ctg5G>

4. Wnioski

Program miał na celu wyświetlanie prędkości obrotowej impulsatora oraz wskazywał kierunek wirowania wykorzystując do tego diody LED. Wykorzystano do tego oprogramowanie Matlab

Simulink, które łączyło się z mikrokontrolerem TEXAS INSTRUMENTS C2000 MCU. Załączone video przedstawia działanie programu, gdzie najpierw impulsator zostaje wykorzystany do wykonania ruchu obrotowego skierowanego w prawo (zapalają się diody LED odpowiadające za ruch prawoskrętny), wyświetla się przy tym proporcjonalna prędkość obrotowa, która zostaje przeliczona z impuls/obr na obr/min, gdzie osiąga wartość 17 obr/min, po odczekaniu w bezruchu 1 sekundy licznik zostaje automatycznie wyzerowany a następnie impulsator zostaje obrócony w lewą stronę, gdzie zapalają się diody odpowiadające ruchowi lewoskrętnemu oraz proporcjonalna wartość prędkości obrotowej a następnie również po czasie 1 sekundy licznik się zeruje. Można tutaj dostrzec pewną wadę impulsatora, który w niektórych przypadkach nie wyzerowuje całkowicie licznika tylko ustawia na nim wartość 2 V co może odbierać za prędkość średnią w momencie nagłej zmiany kierunku obrotu.