Product	AMD	Type/Series	C2000	Appl. Note Nr.	Delta C2000
Issued by	DEN	Author	Marcel Dorti	Release Date	August , 2016
Title	Speed control and position control with MI8 fast-high input pulses counter				

Devices and special tools/equipment

- ✓ Delta C2000 + motor + encoder(if using positioning)
- ✓ EPC Encoder Card (If using positioning)
- ✓ ISPsoft
- ✓ IFD6305 for communication with Delta C2000 frequency converter
- ✓ RS-232 connection cable from DVP to computer USB
- ✓ A PLC with transistor output pulses channel, i.e. Delta DVP28SV

Test setup
N/A

CONTENTS:

1 INTRODUCTION 3
2 CONNECTING HARDWARE 4
3 SETTING THE DVP28SV PLC FOR PULSES.................................... 5
4 SETTING C2000 FOR SPEED MODE WITH MI8 9
5 SETTING MI FOR POSITION MODE 12

1 Introduction

The usage of pulses for setting controls and signal blocks is widely used in PLCs, for a wide range of tasks. Even though, most frequency converters do not count on this resource to receive setpoint from PLCs, this is usually done through analog inputs/outputs or Fieldbus. Delta C2000 has a fast signal receiving digital input, called MI8 high speed input, through this digital input C2000 can receive a setpoint: speed or position. Another advantage is that PLCs usually have few analog outputs built-in, which can be expanded by AI/AO cards. However, with Delta DVP28SV you can assign until 4 channels in order to control external devices such as Delta C2000 drives, through its internal PLCs to set speed and also position from a train of pulses. The MI8 high speed input can count until 33Khz. On this note, we are going to learn how to set this MI8 to work in speed control and in position control.

Figure 1 – DVP28SV sending pulses to 4 C2000 through MI8 high speed input, the internal PLC can define position or speed, for position control an encoder is mandatory (not inserted on the pic).

2 Connecting Hardware

To be able to work with DVP28V pulses and C2000 MI8 high speed input, you need to get a channel of a transistor output and connect the Y0 terminal to the MI8 and the negative terminal C0 connect it to the DCM of the digital inputs, then jump +24V and COM of C2000 terminal block.

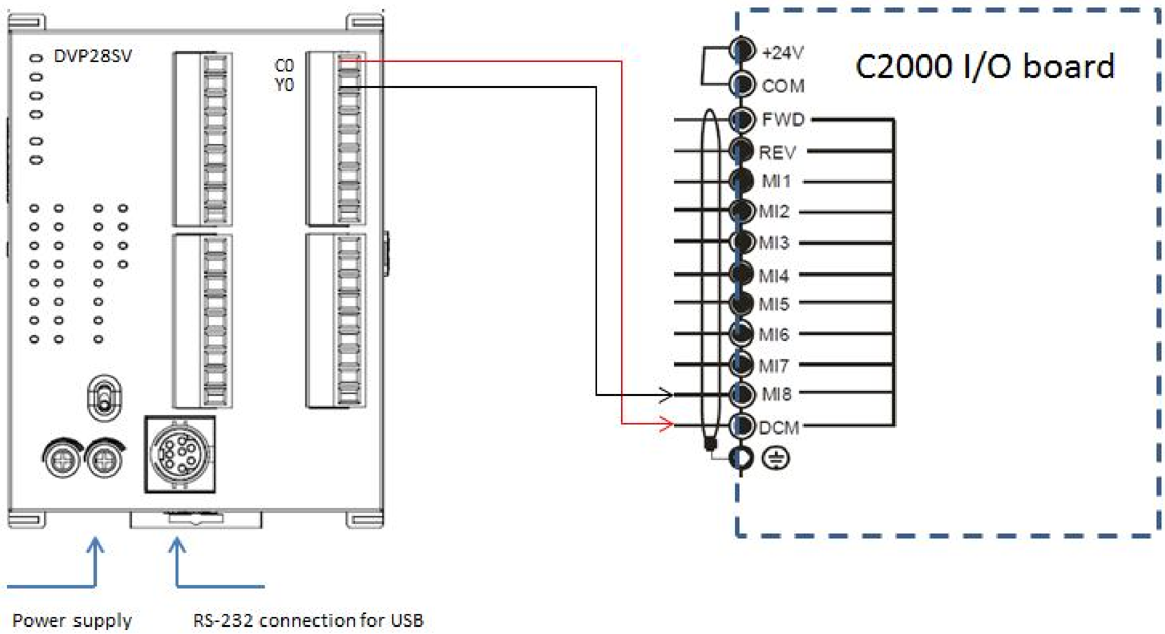

Figure 2 – DVP28SV connected to the MI8 high speed input in C2000

3 Setting the DVP28SV PLC for pulses

Make sure COMMGR is running on your computer with the proper settings, add one connection in one port for the PLC and another one for the C2000. If you do not have it, you will have to download it from www.delta-emea.com and install it in your computer.

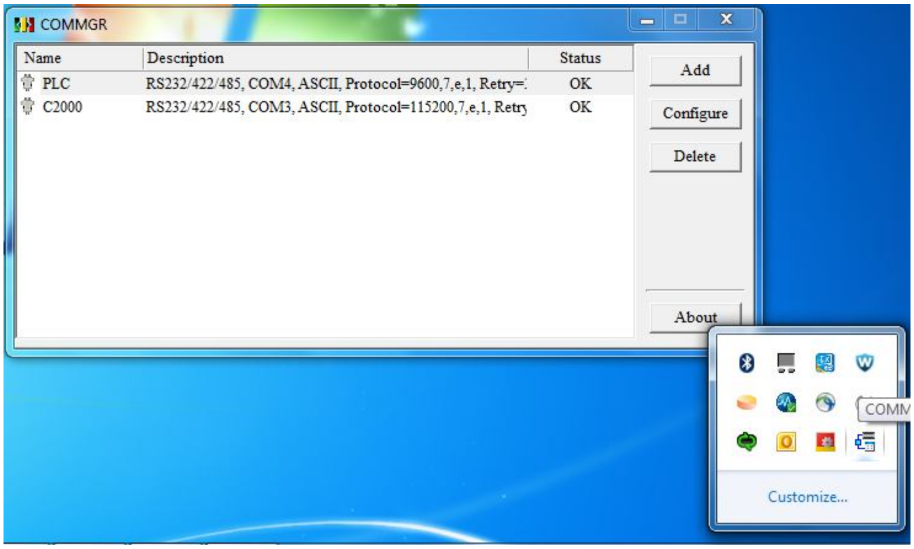

Figure 3 - Checking COMMGR connections. Plug the USB to the device and turn it off to check if they are OK.

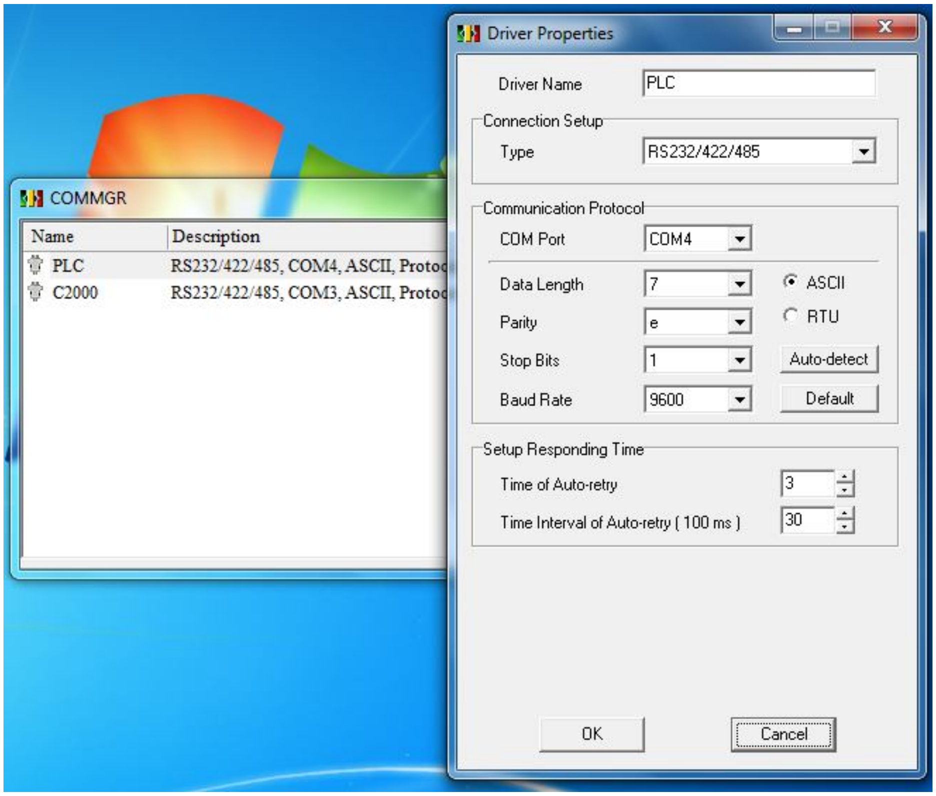

Figure 4 - Insert the correct communication port (you can check that into device manager),

select ASCII or RTU and then Auto-detect).

Now, open ISPSoft, go to communication settings and set your connection with the current address of the DVP28SV, it is initially 1. Go to **Tools -> Communication Settings**. After that, go to **File -> New project,** insert a name and then on the left side bar, go to Programs and add a new program. .

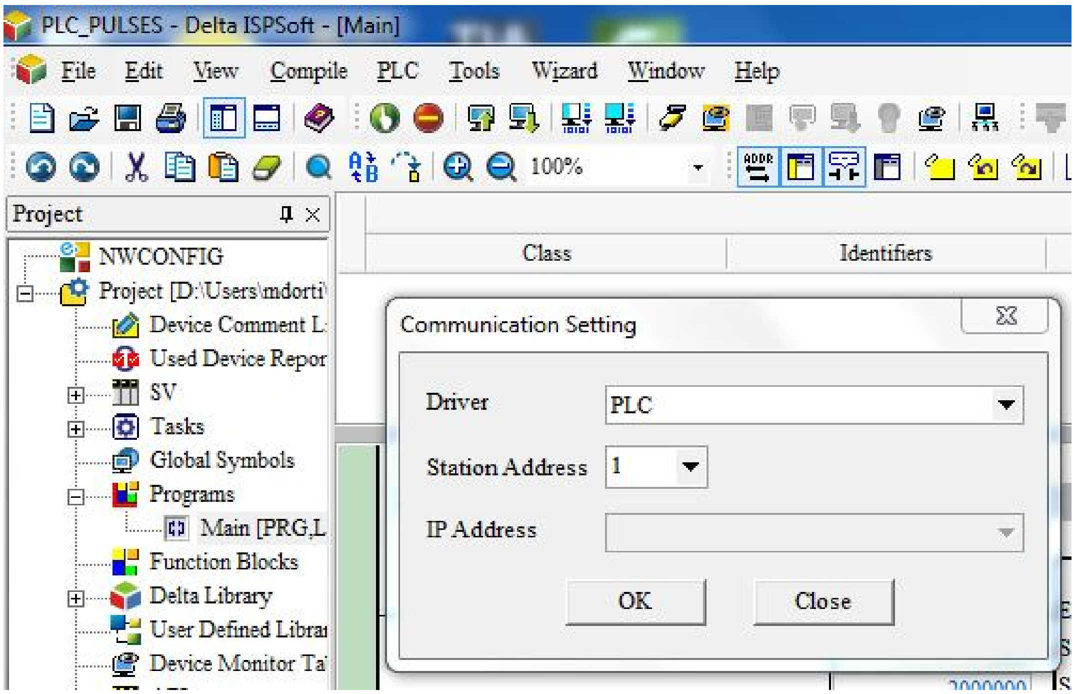

Figure 5 - Setting the communication with PLC.

To run pulses from the pulses output, you will need to add a block called **DPLSY** in your program, go to **API/FB** selection and select the **DPLSY** block.

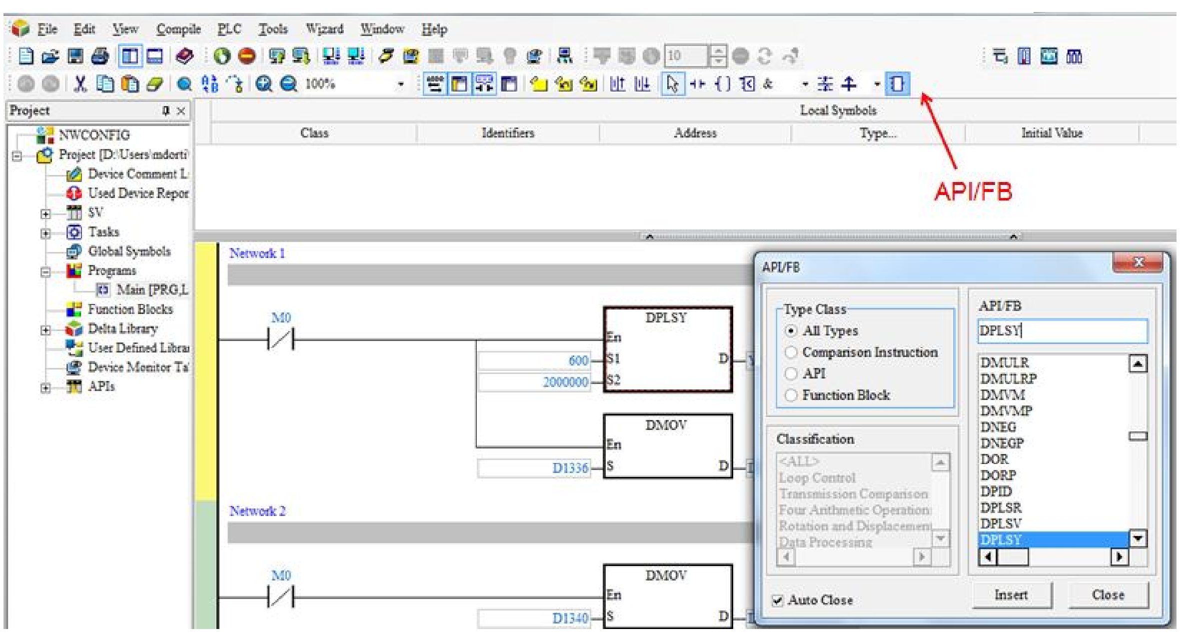

Figure 6 - Block on the API/FB library

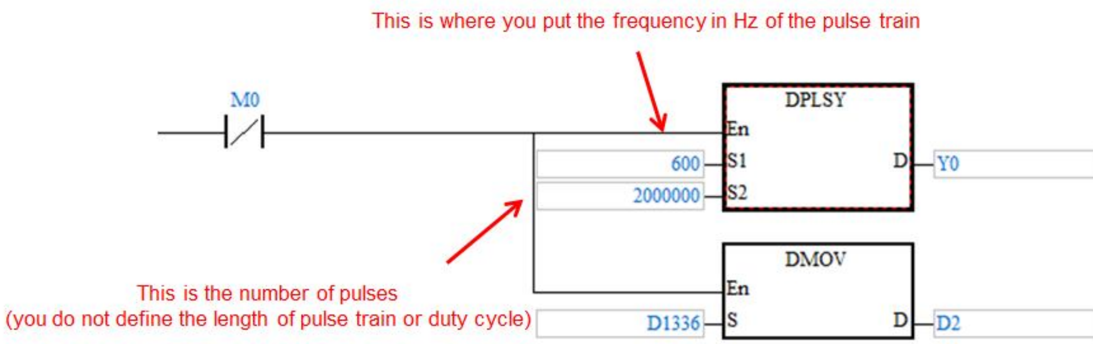

Figure 7 - Block for sending pulses with DVP28SV

On the example above, the pulse will carry 600 Hz and 2000000 pulses, as the period is the 1/f in Hz, we will have a 0.00167 x 2000000 = 3333,33 seconds or 55,56 minutes train of pulses on Y0. D1336 is a special register that stores the number of counts of the pulse.

The control of speed or position in MI8 does not take in consideration the amplitude or length of the pulses, it takes into consideration only frequency, so the setpoint will be kept while there are pulses in frequency on Y0. Now, compile the program with the block, like the one below, **download to PLC** and **Run**.

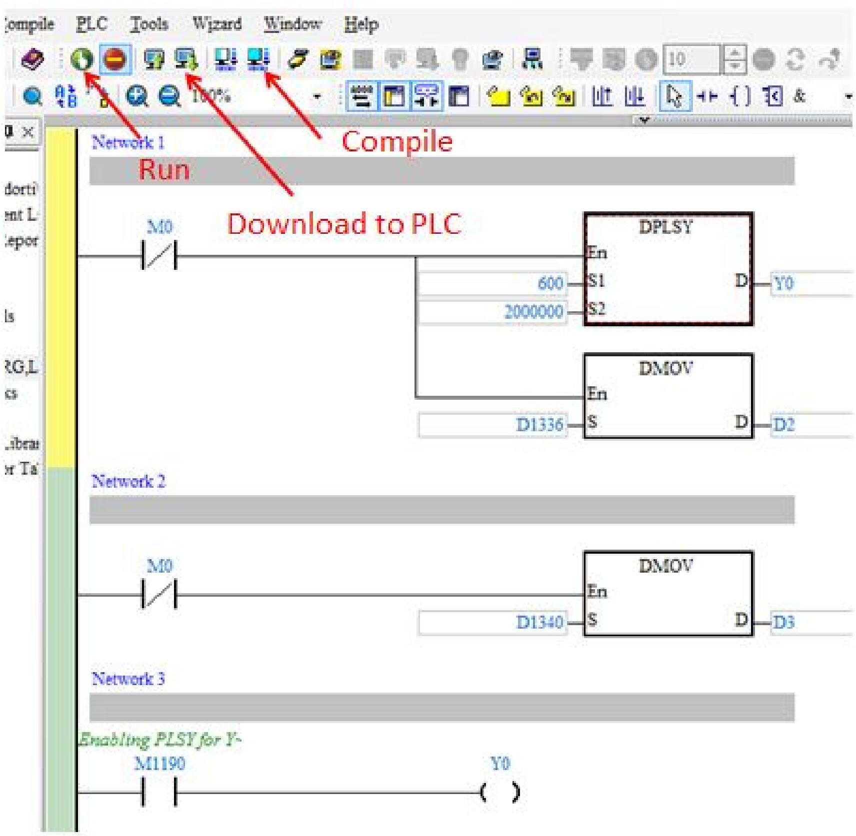

Figure 8 - Small program for PLC pulses with DPLSY

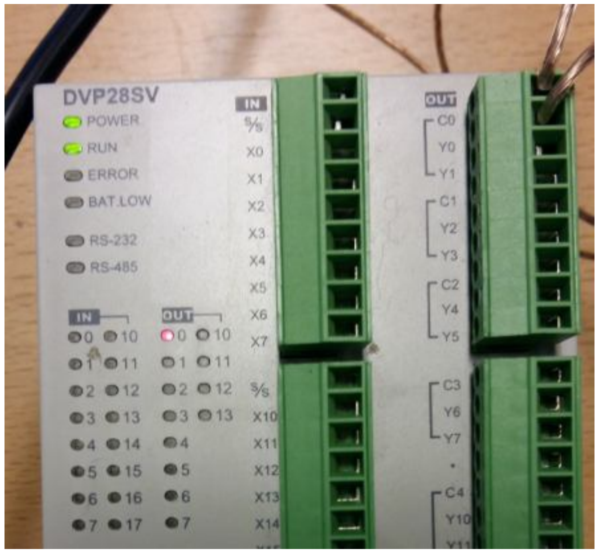

Figure 9 - Check the output LED to check if the pulses are running

4 Setting C2000 for speed mode with MI8

There are 2 registers that handle the count of MI8 pulses received, **D1054**, **D1055**. From Those two registers one is the low byte and the other the high byte of the **double word**. **M1038** is the bit to enable the count from MI8 and **M1039** is the bit to clear the values of the register.

Table 1 - Figure 10 - Important Registers for MI8 count

Register	Function
D1054	Low byte of pulse double word
D1055	High byte of pulse double word
M1038	Bit to start the count
M1039	Bit to clear the registers of D1054,D1055

Open ISPsoft again, start a new project and define C2000 as the PLC, open a new program and insert the following move instructions, then you will be able to check the pulses which come from the PLC we had just set on the previous chapter. Go to Communication settings and define the C2000 connection of COMMGR, insert the C2000 PLC (2 – default). Compile, download and RUN the program.

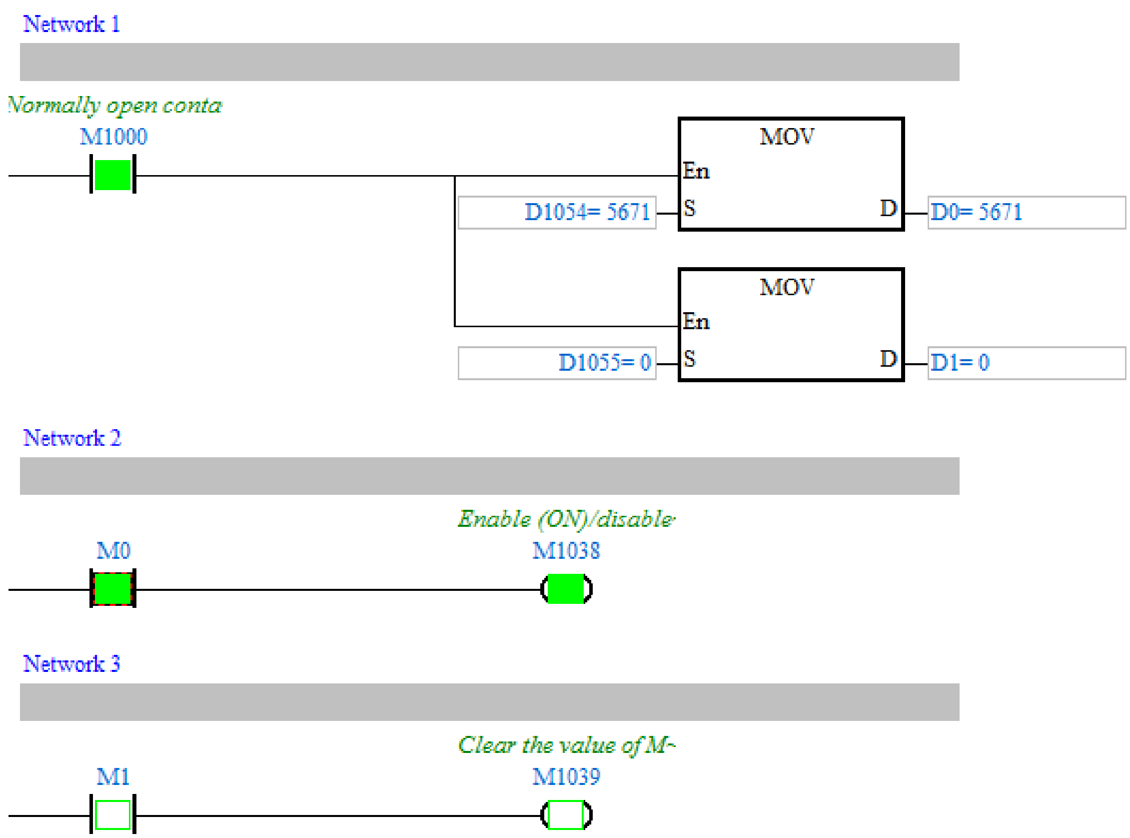

Figure 10 - receiving pulses from MI8

Now, to use the pulse command, the frequency pulse command must be converted into a speed command, there are other registers to complete this equation, to convert frequency into rotational speed, use the following equation:

$$D1056 = \frac{frequency\ of\ pulses\ in\ Hz}{D1057} \cdot 10^{D1059}$$

D1056 is the desired rotational speed to be inserted as frequency command in the drive. Frequency of pulses, is the frequency which should be assigned entry S1 **of DPLSY block** in the PLC. **D1059** is the number of decimal places in the calculation, use 1 always not to have any problem with the calculation. **D1058** is the interval of pulses you want the PLC to calculate the result for 1056. **D1057** is a factor (speed ratio) in the equation to achieve the final result which will be inserted in the FREQ block, multiply by 100 the number. For example, If you want 60 Hz speed in the drive, the result of **D1056** should be equal to 6000.

Table 2 - Important Register for pulse - frequency conversion

Register	Function
D1058	interval of calculations in ms (should be around 1000)
D1059	Decimal place, should be equal to 1
D1057	Speed Ratio, a factor in the equation
D1059	Frequency in Hz, x 100

Example 1 – In order to achieve a speed of 60Hz in D1056, which should be the pulses frequency of the DPLSY block on the PLC? Considering D107 = 1 and D1059 also 1. How much time would this train of pulses take to achieve 0 frequency? (End of train of pulses)

$$6000 = \frac{frequency\ of\ pulses\ in\ Hz}{1} \cdot 10^{1}$$

$$6000\, x \frac{1}{10^{1}} = frequencyof\ pulses\ in\ Hz = 600\ Hz$$

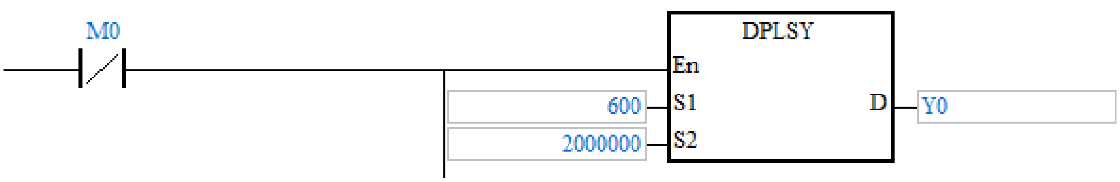

Figure 11 - Block with calculated frequency for speed

Considering the number of 2000000 pulses, and a frequency of 600HZ, period = 1/frequency = 0,00167 s x 2000000 = 3333,33 seconds or 55,55 minutes for the train of pulses to come to an end.

To achieve those values in the equation, you should use **MOV** instructions of the library like the example below, the value of **M1056** will change according to the values set.

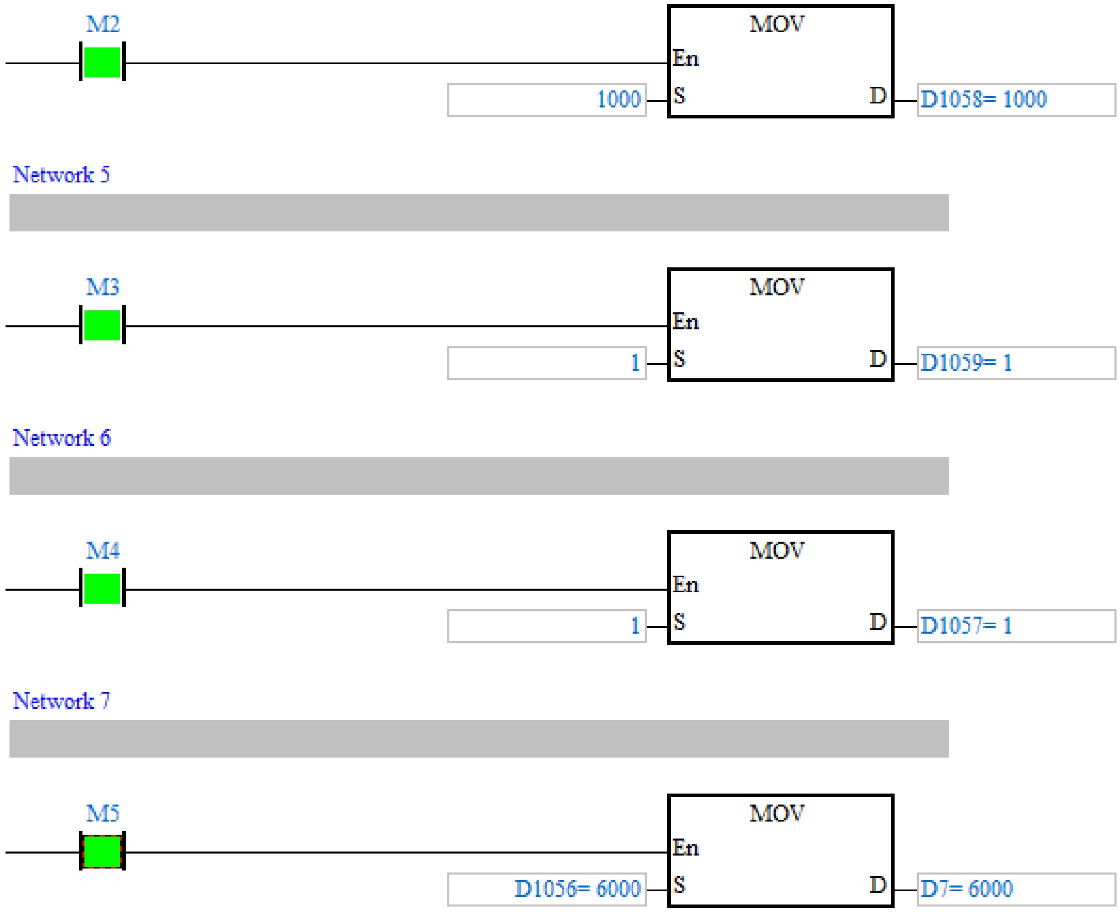

Figure 12 - Elements of the equation to convert pulses into frequency

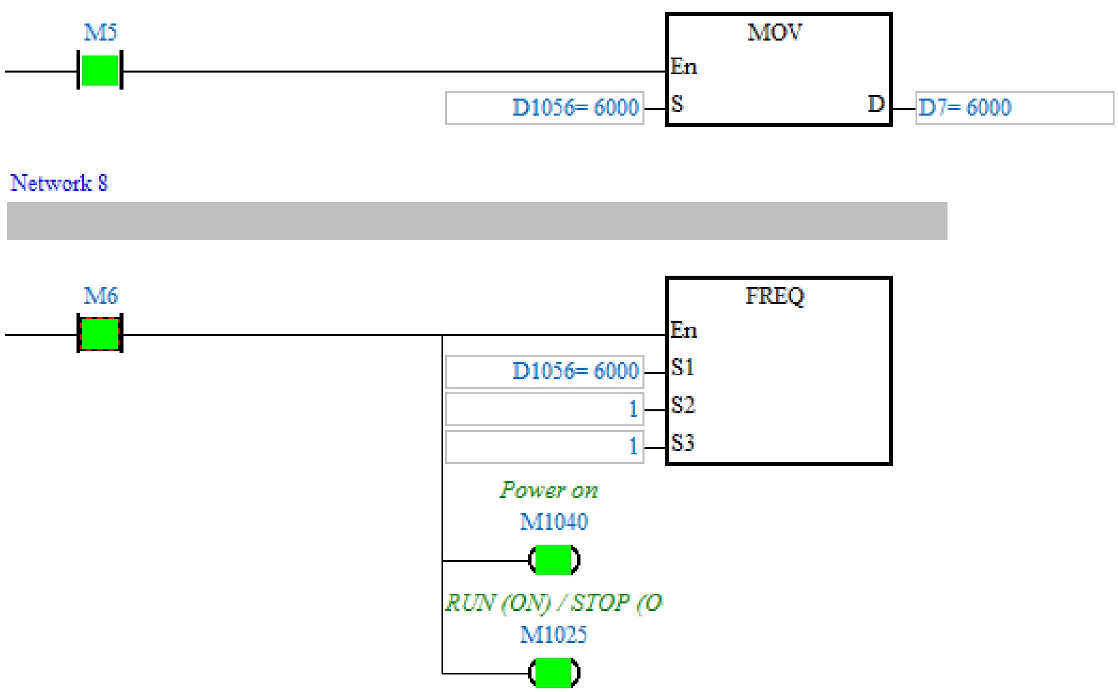

Figure 13 - Insert the value of D1056 in a FREQ block, set M1040 and M1023 to enable the drive. Changing the frequency of the pulses will change the frequency according to the equation mentioned above.

5 Setting MI for position mode

You can use the same number in M1056 register in order to achieve positioning function with the block DPOS, naturally the number of pulses you set to the position will be different from the encoder pulses (00-04 = 21) you need to calculate a factor between the encoder pulses of your application and the pulses of the frequency sent to the command.

The position block DPOS just works if not enabled at the same time of M1040 (drive power-on), because of that you need to implement a timer routine to get the position function accomplished. To start it, you should set a network to change to position mode, D1060 = 1, in speed, D1060 = 0.

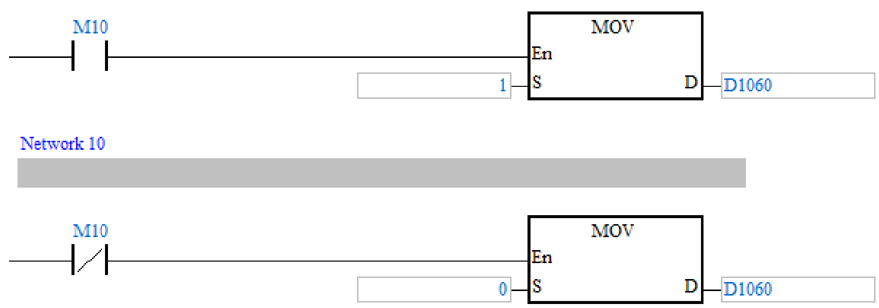

Figure 14 - Setting D1060 register to position, when M10 is enabled, D1060 = 1

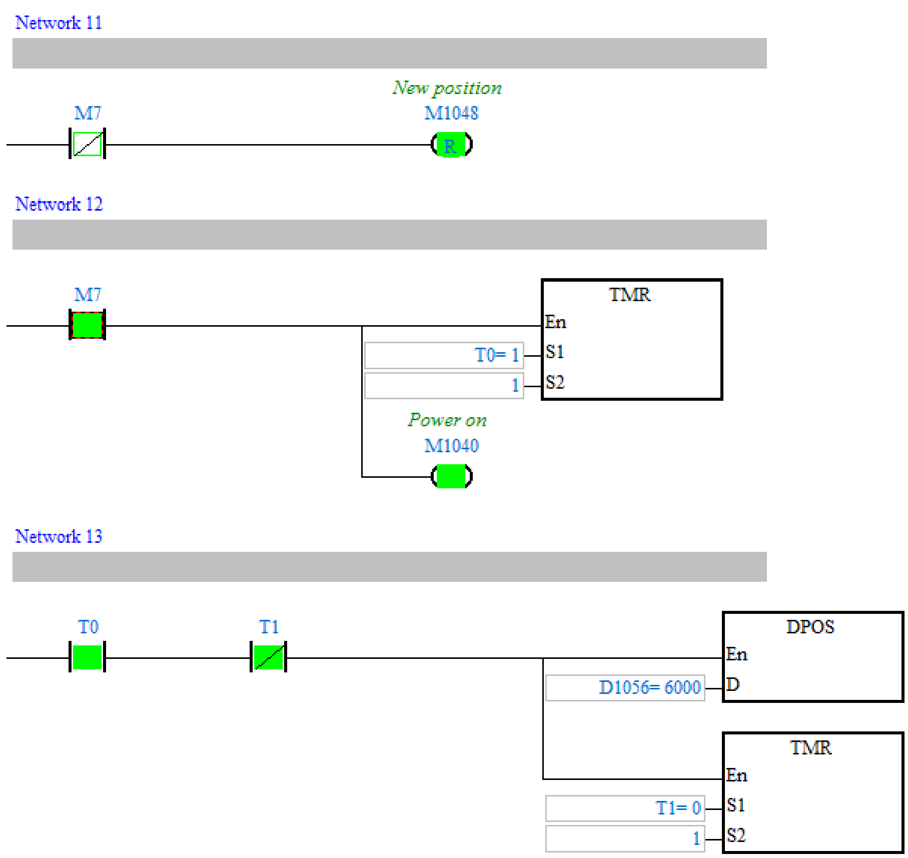

Figure 15 - Positioning routine, when D1060 = 1, and M7 is enabled drive is

powered and T0 time after position assigned to DPOS from D1056

Figure 16 - After T1 time is up, T1 will be enabled and M1048 set for new position

You can use the same equation for D1056 to calculate the number of pulses, just be aware they are not the same number of encoder pulses.