Preliminary

Reference

SJ700 Easy Sequence Function Outline (provisional)

Ver.0.03

This specification can be changed without notice.

● Easy Sequence Function - Feature

Preliminary

Reference

<Feature>

1. Allow users to customize their inverters.
2. Control of inverter is realized by easy programming.
3. Utilizing calculation functions, any signal can be output.

<Notice>

Motion commands used for AD servo, are not implemented.

● Easy Sequence Function - Spec (1)

Preliminary

Reference

Item		Description
Language Spec.	Language type	BASIC Like
	Supported Device	IBM PC/AT compatible (Windows 98SE, 2000, XP) Inverter: RS-422 port common
	Memory area	512 steps or 4K byte (Smaller of these) Program is stored in internal of inverter.
	Programming environment	Editor （Windows） Display （Windows） Grammar check （Windows） Program download/upload, All clear
	Executable format	Interpreter 2.0ms/command (Sub routine supported. 8 nested)
I/O function	External digital contact input	Contact signal / Open collector signal input (Internal DC24V power supply available) Program RUN command （FW terminal is reserved for this function） Alarm reset （Terminal #8 is reserved for this function） 7 general purpose input registers （X(00)～X(06)）
	External output	5 general purpose output registers （Y(00)～Y(04)） 1 Relay output (alarm)
	External analog input	3 input （XA(0)～XA(2)） O input （0～10V）, OI input （4～20mA）, O2 input （±10V）

● Easy Sequence Function - Spec (2)

Preliminary

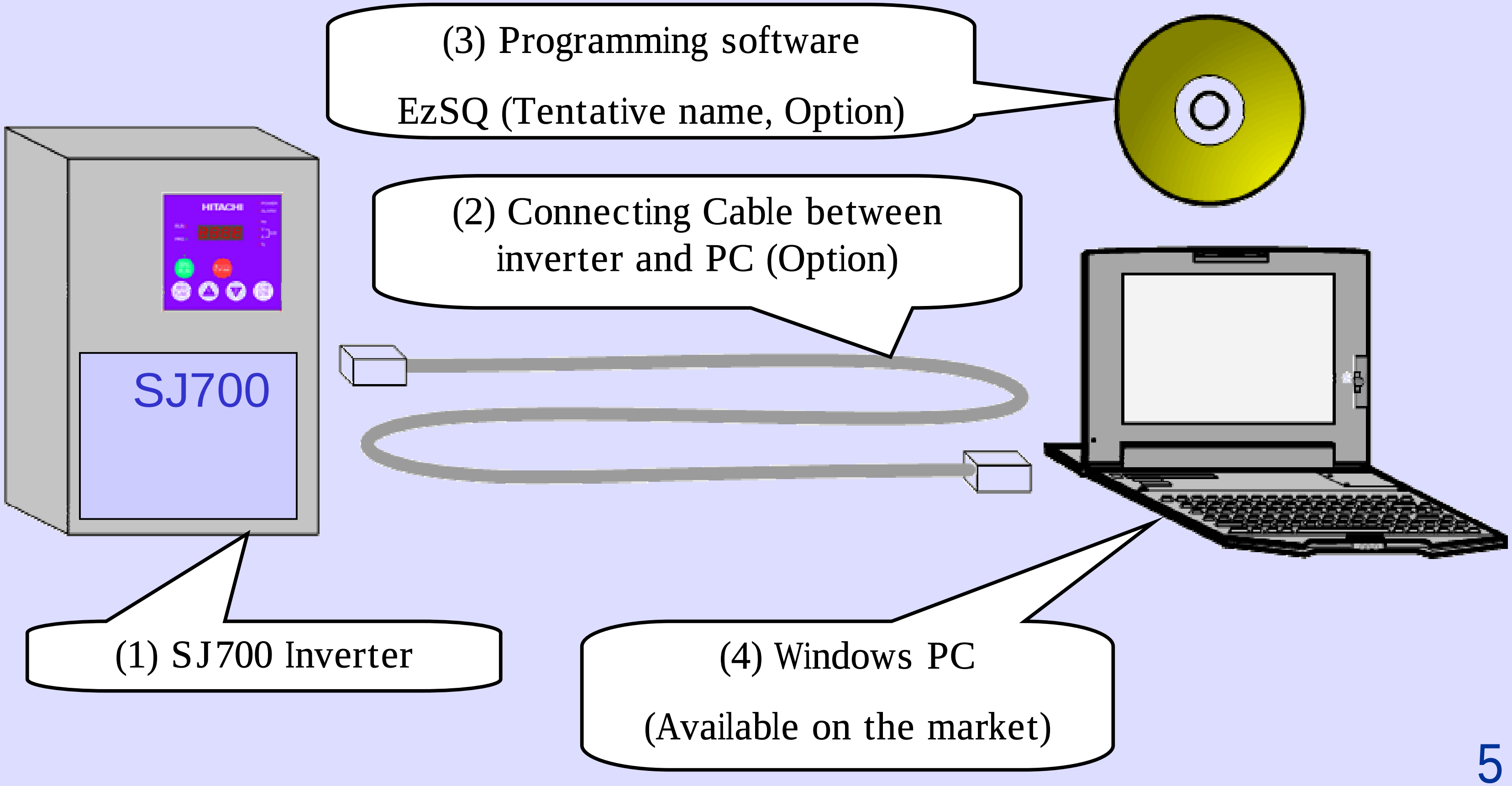
Reference

Item			Description	
Reserved word	Variable	User	U(00)～U(31)（32 registers）	
		Timer	TD(0)～TD(7)（8 registers）	
		Setting	SET-Freq（Set frequency）	
			ACCEL(Acceleration time)	DECEL（Deceleration time）
		Monitor	FM（Output frequency）	Iout（Output current）
			Dir（Rotative direction）	PID-FB（PID feedback）
			F-CNV（Converted freq）	TRQ（Output torque）
			ERR CNT（Trip number）	ERR(0)～ERR(6)（7 point, Trip factor）
	Command	(1) Programmable flow control Loop (for) Unconditional jump（goto） Conditional jump（if, ifs then, select case, until, while） Time control（wait） Sub routine（call, sub） Read / write of variable（load, save） Others（entry, end, stop cont, inc, dec, …etc） (2) I/O control（Bit input, Word input, Bit output, Word output） (3) Internal register control（Inverter control function, on delay, off delay） (4) Inverter parameter setting		
Operator		Mathematical	+, -, *, /, substitution, mod, abs	
		Logical	or, and, xor, not	
		Conditional	=, <, <=, >, >=, <>	

● Requirements for Easy Sequence Function

Preliminary
Reference

(1) SJ700 Inverter
(2) Connecting Cable between inverter and PC（Option）
: Insert this cable into the connector after the OPE-S is removed.
(3) Programming software EzSQ（Tentative name, option）
: Similar environment are provided for the operation as AD-servo setup software of AHF-P02.
(4) Windows PC（Available on the market）


(3) Programming software
EzSQ (Tentative name, Option)
HITACHI
SJ700
(2) Connecting Cable between inverter and PC (Option)
(1) SJ700 Inverter
(4) Windows PC
(Available on the market)

● How to perform Easy Sequence Operation

Preliminary
Reference

(1) Inverter operates in normal way unless easy sequence is written on an SJ700 inverter by the setup software.

(2) If easy sequence is written on the SJ700 inverter, the inverter shows operation mode switch parameter and program clear parameter.

(3) Intelligent input terminals assignment is changed by the operation mode switch parameter.

- Normal mode:
  The intelligent I/O terminal is assigned according to parameters of C001-C008 and C021-C025.
- Easy sequence mode:
  [RUN], [RS] and general purpose I/O terminals （X(00)～X(06), Y(00)～Y(04)）

Preliminary

Reference

SJ700 Easy Sequence Function Application Notes

Application example [1] Multi-speed

Application example [2] Crain (Hoist)

Application example [3] Fan and Pump

Application example [4] Sequential control
(holding action, time delay action)

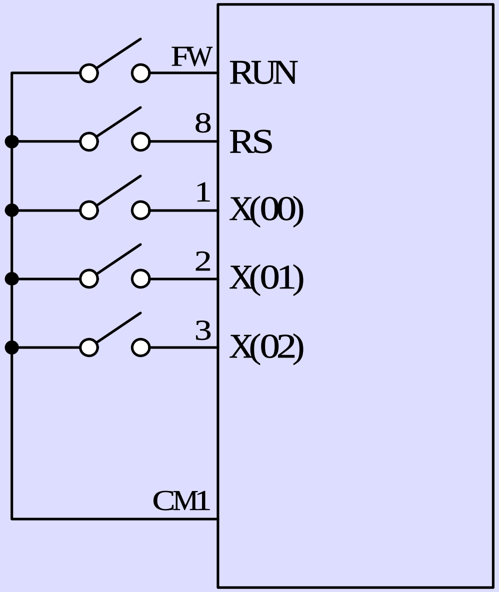
Application example [1] - Multi-speed

Preliminary

Reference

- Multi-speed is implemented using simple sequence operation in this example
- Same function is achieved as multi speed function in normal mode using terminals assigned [CF1] to [CF4].
- Time control function of sequence operation are utilized in order to get a delay from acutual input for a frequency swicth.

< Wiring example >



< Outline of operation >

General-purpose inputs are used to select the multi -speed.
X (00) and X (02) are used.
0 to 4 of 0 to 7 multi-speeds are used.
Multi speed 0 is stop.
Multi speed 1 is 10Hz and forward direction.
Multi speed 2 is 60Hz and forward direction.
Multi speed 3 is 10Hz and reverse direction.
Multi speed 4 is 60Hz and reverse direction.
5 to 7 speed inputs are neglected. Maintain current motion.
It takes 1 second as delay time to actually switch frequency after multi-speeds input
For the case of 0 speed, delay time does not exist.

Application example [1] - Multi-speed

Preliminary

Reference

<Program example>

```
[Program area]
               entry
 LOOP:         U(00)=         Xw      and 7
               select  U(00)
               case    0
                              stop
               case    1
                              wait         1.0
                              turn on FW
                              SET-Freq     =1000
               case    2
                              wait         1.0
                              turn on FW
                              SET-Freq     =6000
               case    3
                              wait         1.0
                              turn on RV
                              SET-Freq     =1000
               case    4
                              wait         1.0
                              turn on RV
                              SET-Freq     =6000
               case    else
               end select
               go to          LOOP
               end
[Data area]
  ACCEL=100         'Accel Time 1s'
  DECEL=100         'Decel Time 1s'
```

Start of program

Word X(00) to X(02) of all general-purpose input terminals are put into U(00)

Wait 1 second before frequency is modified

Depending on the X(00) to X(02), freqency and rotative direction are selected

Repeat from the label named LOOP.

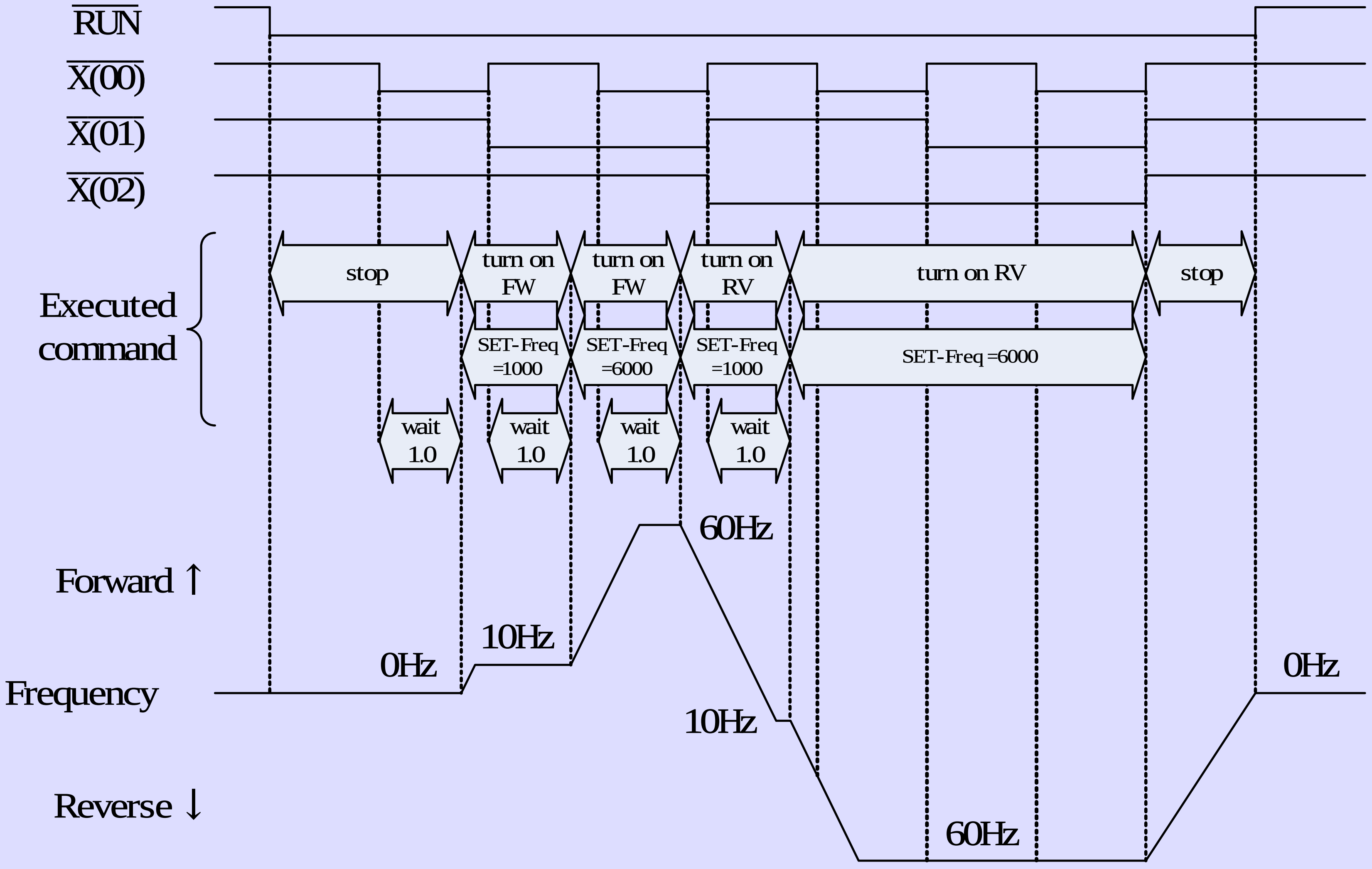
Program end

Application example [1] - Multi-speed

Preliminary

Reference

<Operation example>



Application example [2] - Crane (Hoist)

Preliminary

Reference

- Control of crane are implemented using simple sequence operation.
- Example of brake control and automatic speed increasing.
- General-purpose input/output is assigned as follows:

Ascent signal	X(00)
Descent signal	X(01)
Automatic speed increase signal	X(02)
Brake open verification signal	X(03)
Brake open signal	Y(00)

< Operation outline >

1. Wit until Ascent(Descent) signal becomes ON.
2. When Ascent(Descent) signal becomes ON, it accelerates acceleration creep speed.
3. When it arrives acceleration creep speed, it opens the brake.
4. When brake open signal is turned to ON, it accelerates to the setting speed.
5. When Ascent(Descent) signal becomes OFF, it decelerates to the deceleration creep speed.
6. When it arrives deceleration creep speed, it closes the brake.
7. If the brake open verification signal becomes off, it decelerates and stops
8. During the operation at the setting frequency with automatic speed increase signal is ON , if output torque is not enough, it increases frequency depending on the torque.

Application example [2] - Crane (Hoist)

Preliminary

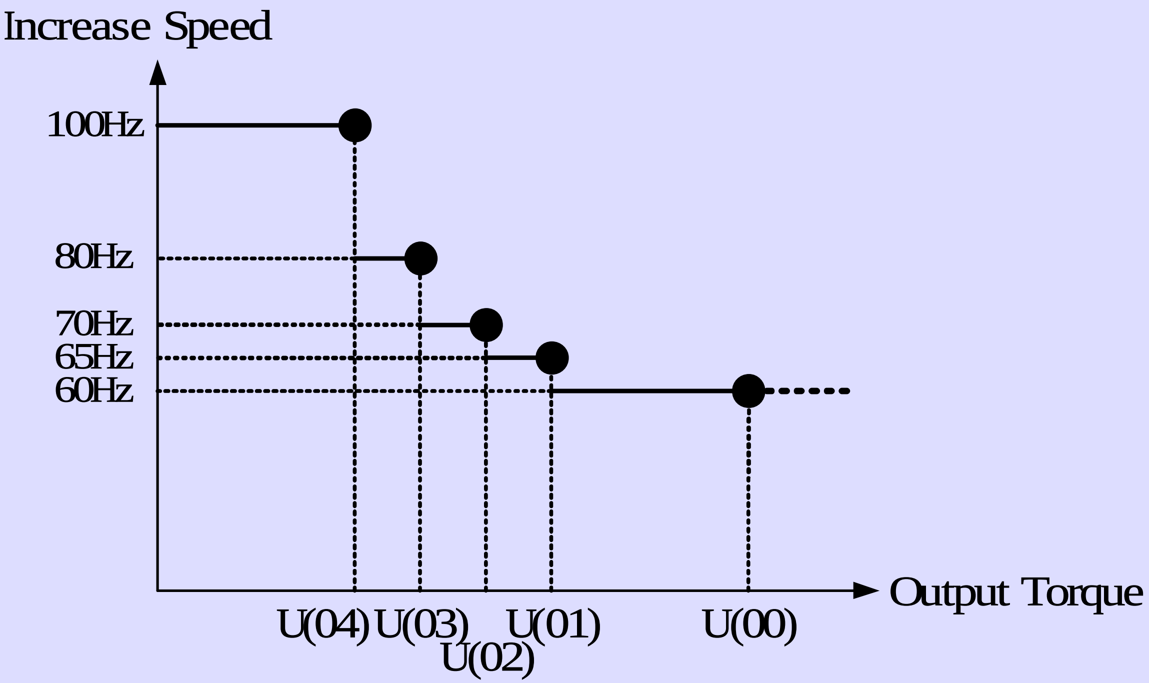
Reference

< Operation outline >

Automatic speed increase operation is perfomed when it is operated at setting frequency (60Hz) and automatic speed increase signal becomes ON .

Following conditions are implemented with the simple seaquence to increase speed (example).

if output torque (TRQ) is U(00) to U(01), SET-Freq=60Hz (base frequency)
if output torque (TRQ) is U(01) to U(02), SET-Freq=65Hz
if output torque (TRQ) is U(02) to U(03), SET-Freq=70Hz
if output torque (TRQ) is U(03) to U(04), SET-Freq=80Hz
if output torque (TRQ) is smaller than U(04), SET-Freq=100Hz (max frequency)

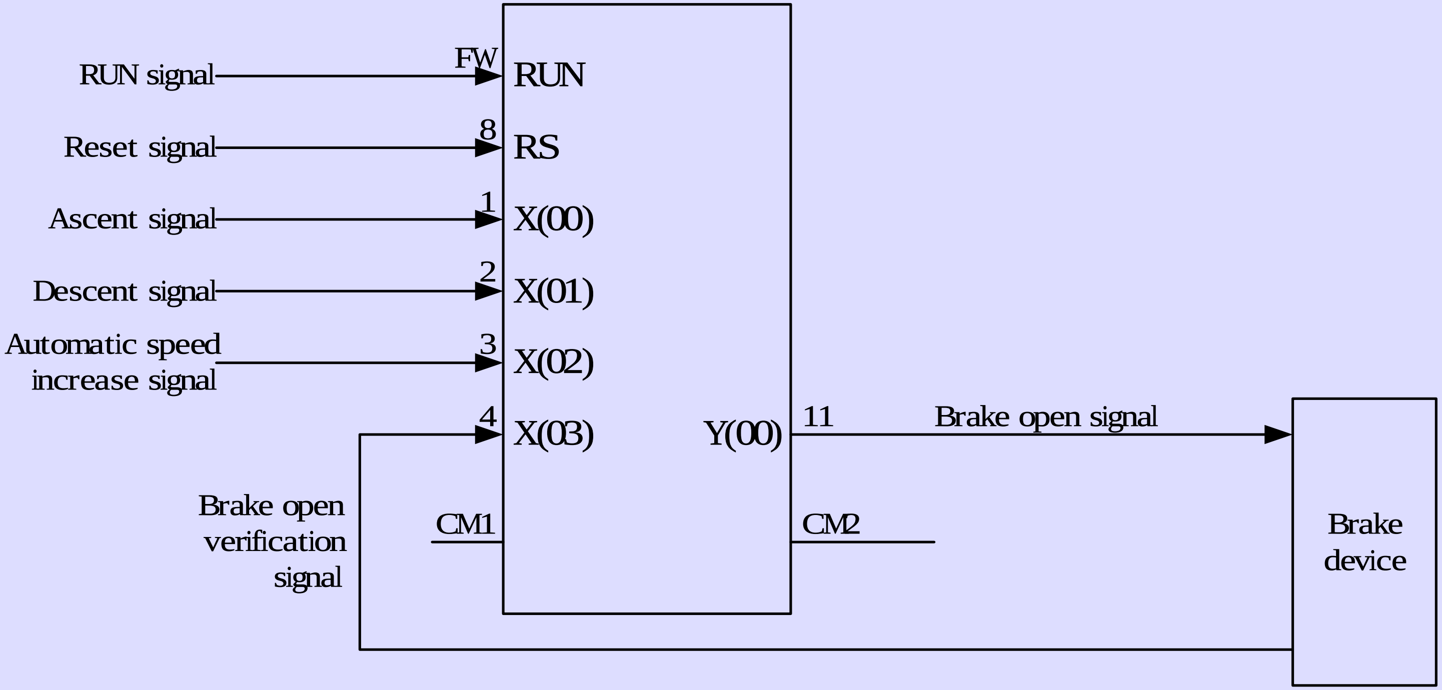


Application example [2] - Crane (Hoist)

Preliminary

Reference

<Wirign example>

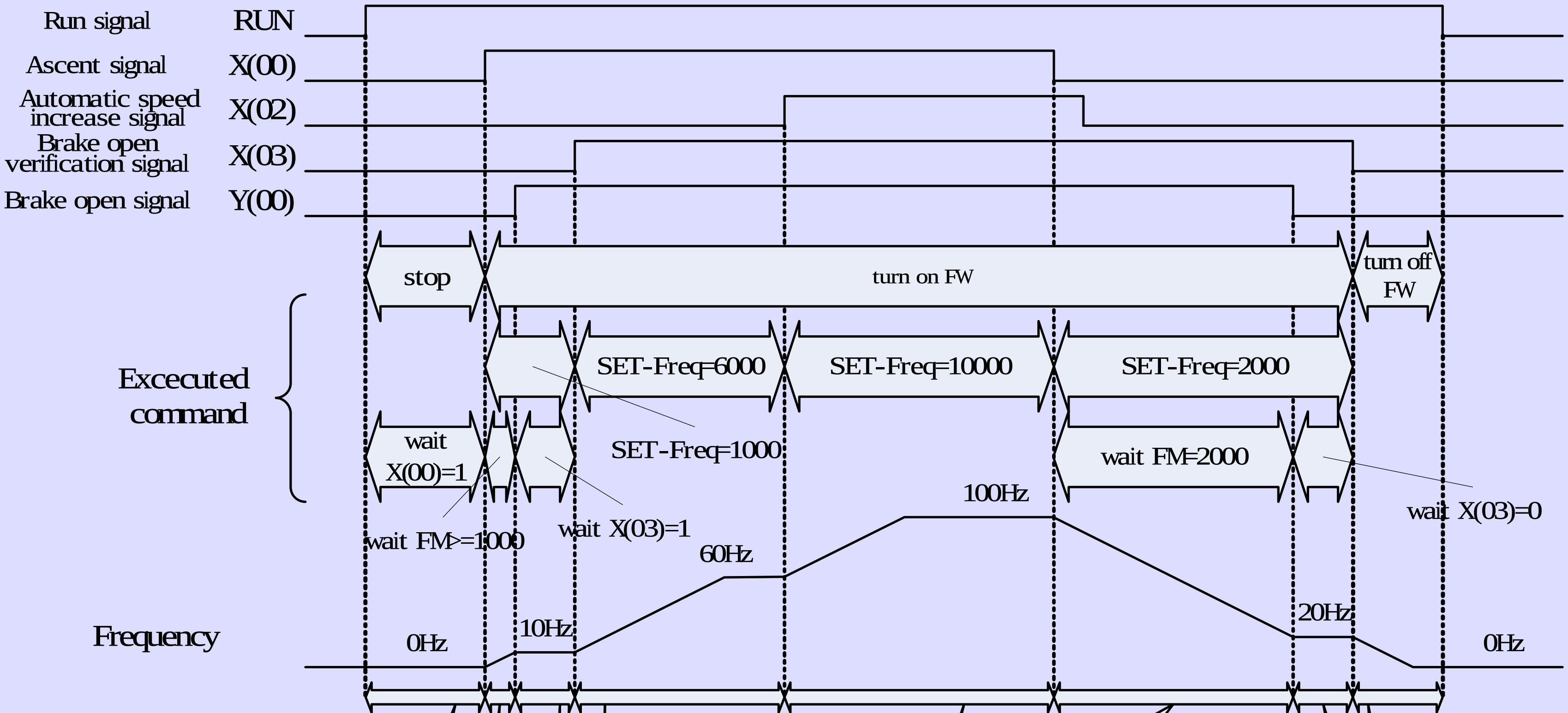


Application example [2] - Crane (Hoist)

Preliminary

Reference

<Operation example>



1. Waiting for ascent (descent) signal.

2. If ascent (descent) signal is ON, accelerate to acceleration creep speed (10Hz)

3. If acceleration creep speed is reached, turn on the brake open signal and wait for the brake verification signal turning ON.

4. If brake verification signal becomes ON, it accelerates to 60Hz.

5. If ascent (descent) signal becomes OFF, it decelerates to deceleration creep speed (20Hz).

6. Deceleration creep speed is reached, turn OFF the brake open siganal and wait for the brake verification signal turning OFF.

7. If brake verification signal becomes OFF, it decelerates to stop

8. If automatic speed increase signal is turned ON, it increases to the frequency decieded by torque.

Application Exapmle [3] - Fan and Pump

Preliminary

Reference

- Perform a constant pressure control (PID), operation number control and sleep status control for fan and pump.
- Following signals are assigned to general purpose output terminal

Reference value input	XA(0)	RUN(Forward) signal	X(00)
Feedback value input	XA(1)	PID comparison output signal	Y(00)

- PID feedback calculation is also performed in the simple sequence

<Outline of operation>

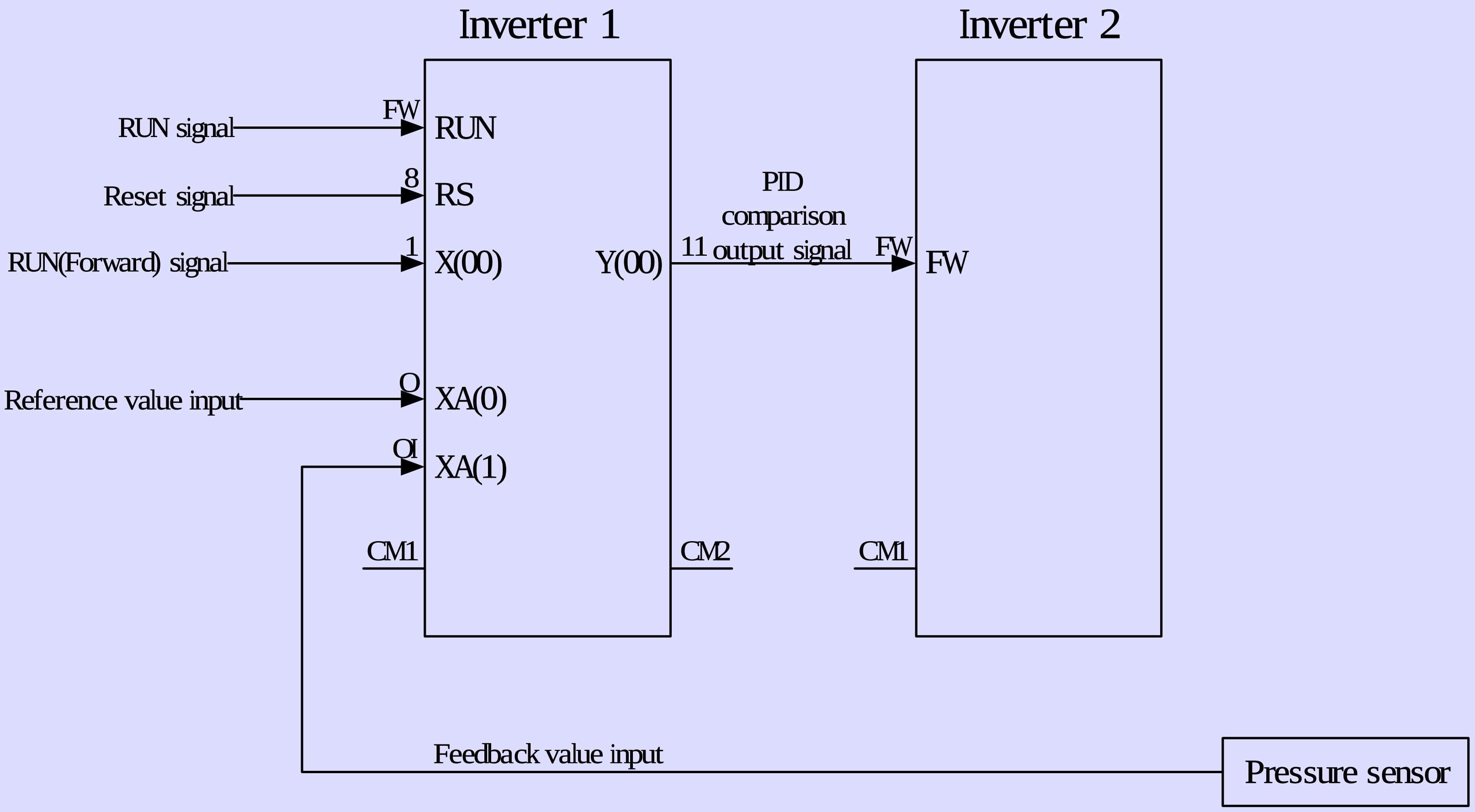
1. Wait for the RUN(Forward) signal is turned on.
2. If RUN(Forward) signal is turned on, start the operation. PID control is performed using reference and feedback values. This calculation will be continued until No.6. Turn on the PID comparison output signal
3. If feedback value becomes larger than OFF level of PID comparison output, turn OFF the PID comparison output signal
4. If feedback value becomes lower than ON level of PID comparison output, turn ON the PID comparison output signal
5. Even if RUN(Forward) signal is ON, if feedback value becomes less than sleep level it decelerate to stop and becomes sleep status and turn off the PID comparison output signal
6. If feedback value becomes more than sleep level, operation starts. and turn ON the PID comparison output signal
7. Feedback value becomes more than OFF level of PID comparison output signal, turn off the PID comparison output signal
8. Feedback value becomes less than ON level of PID comparison output signal, turn ON the PID comparison output signal
9. If RUN (Forward) signal is OFF, turn OFF the PID comparison output signal and decelerate to stop.

Application Example [3] - Fan and Pump

Preliminary

Reference

<Wiring example>

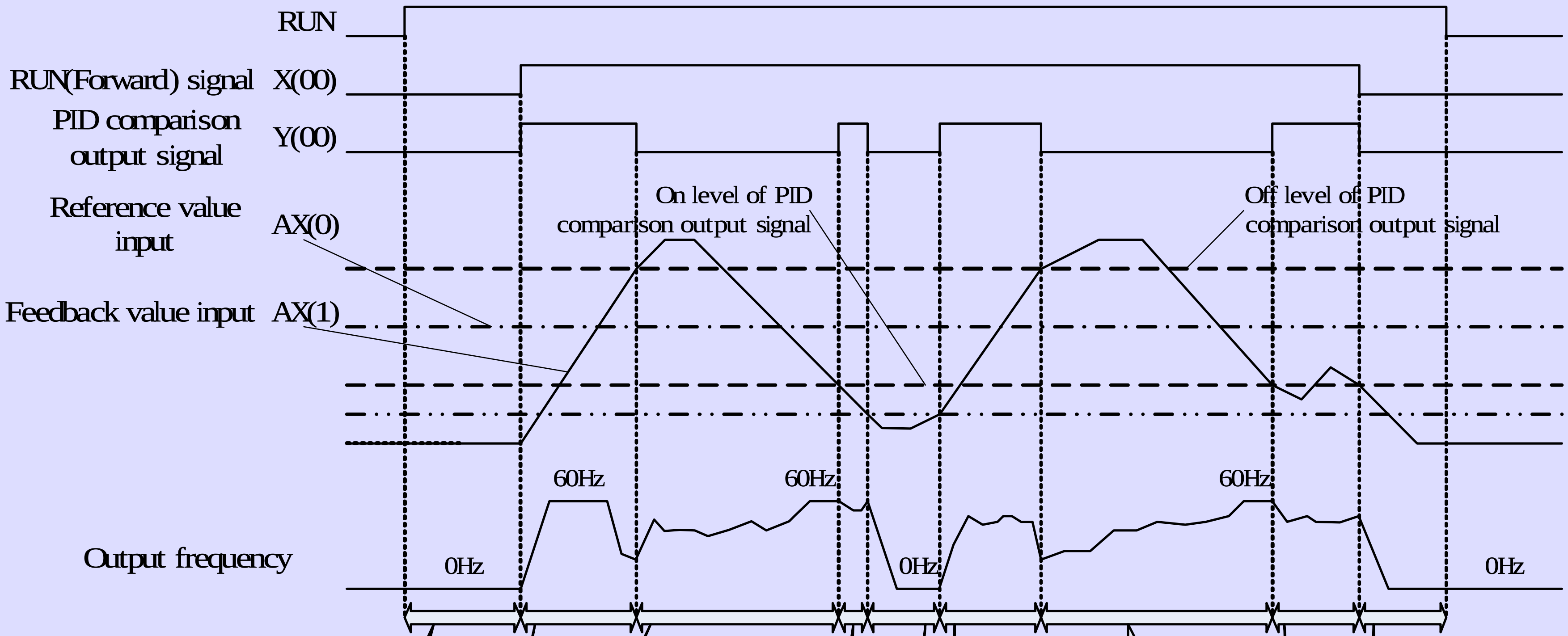


Application Example [3] - Fan and Pump

Preliminary

Reference

<Operation example>



1. Waiting for RUN(Forward) signal turning ON without operation.

2. If RUN(Forward) signal becomes on, perform PID calculation and starts operation and turn on the PID comparison output signal.

3. If feedback value becomes more than Off level of PID comparison output signal, turn off the PID comparison output signal.

4. If feedback value becomes less than on level of PID comparison output signal, turn on the PID comparison output signal.

5. If feedback value becomes less than sleep level, it decelerates to stop.

6. If feedback value becomes more than sleep level, it starts operation

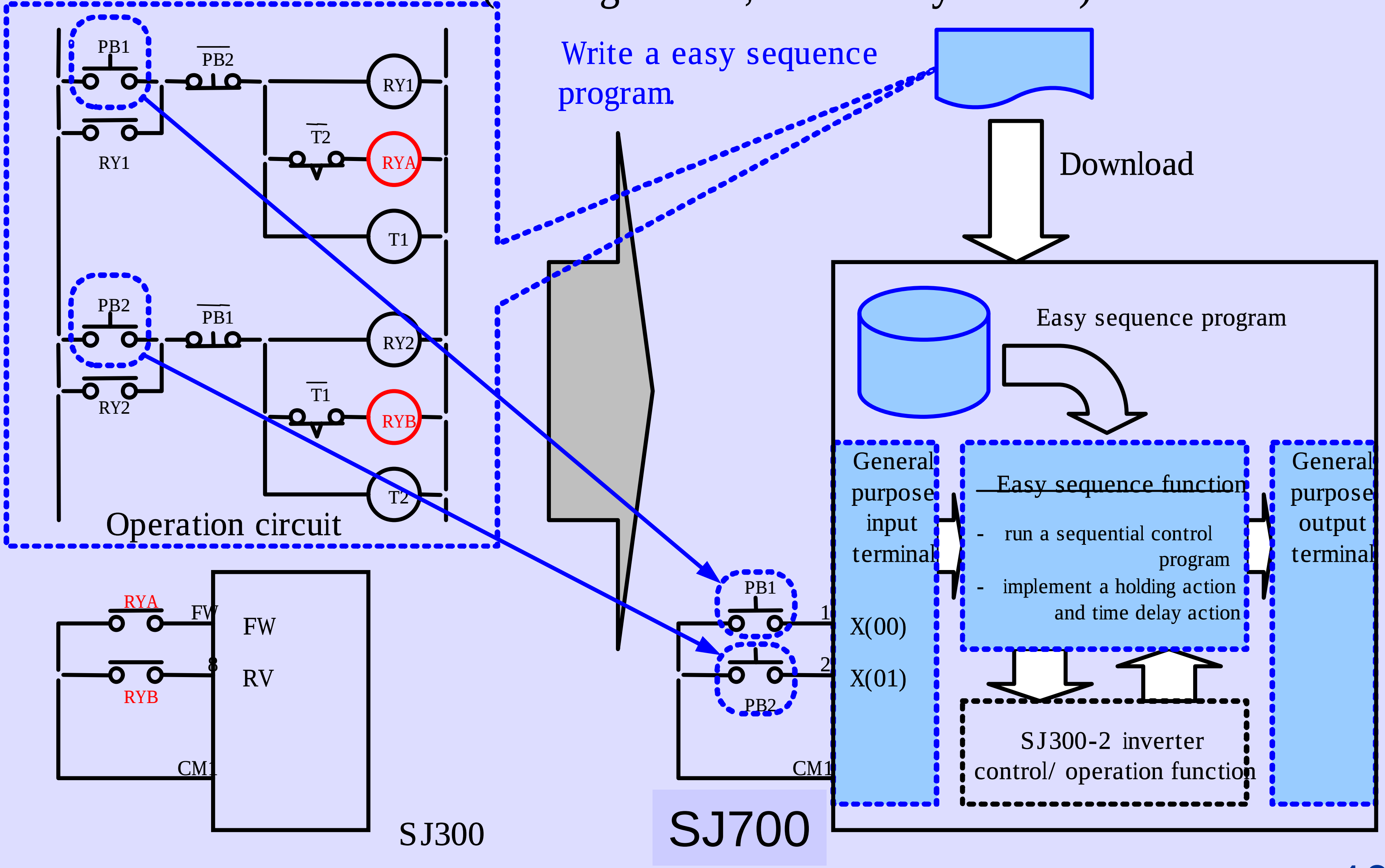
7. If feedback value becomes more than off level of PID comparison output signal, turn off PID comparison output signal .

8. If feedback value becomes less than on level of PID comparison output signal, turn on PID comparison output signal.

9. If RUN(Forward) becomes off, it decelerates to stop.

Application Exapmle [4] - Sequential control (holding action, time delay action)

Preliminary
Reference

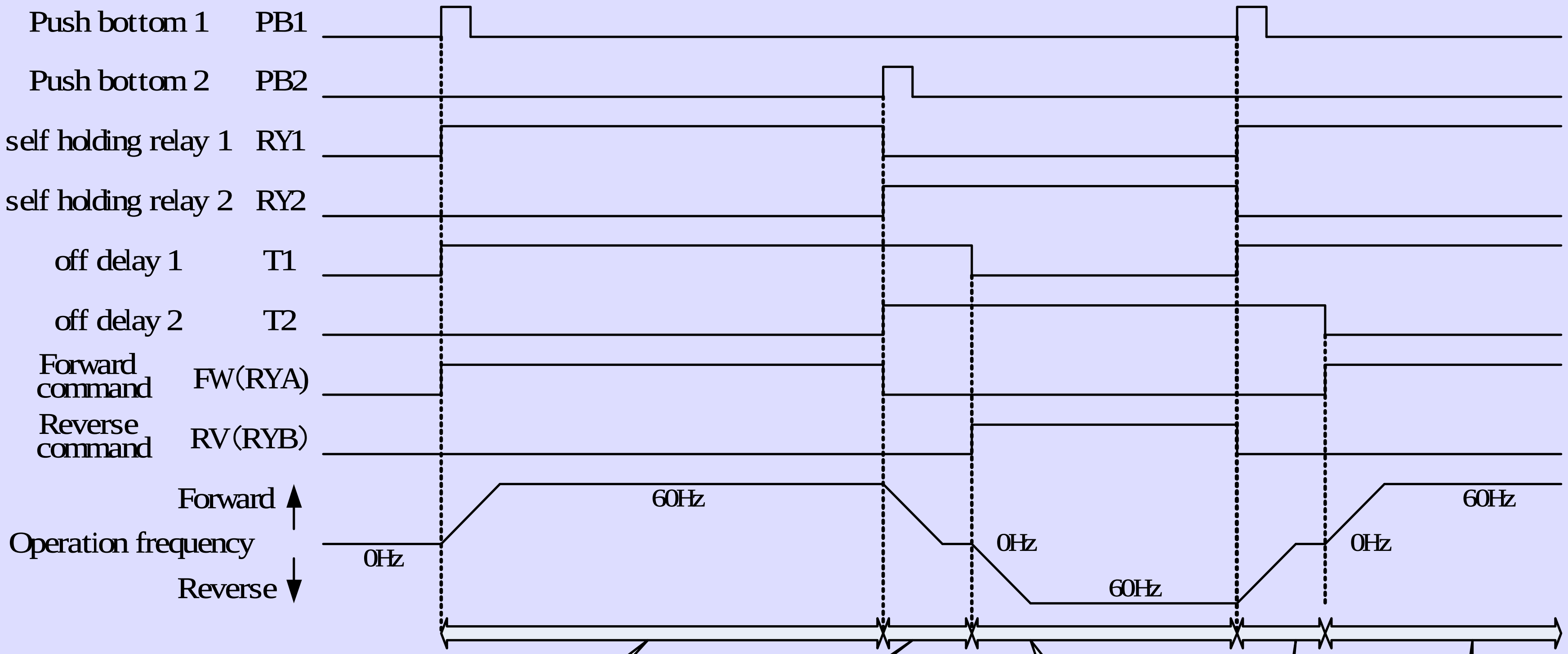

PB1
PB2
RY1
T2
RYA
T1
RY1
PB2
PB1
RY2
T1
RYB
T2
RY2
Operation circuit
Write a easy sequence program.
Download
Easy sequence program
General purpose input terminal
Easy sequence function
- run a sequential control program
- implement a holding action and time delay action
General purpose output terminal
RYA
FW
FW
RV
RYB
CM1
SJ300
PB1
1
X(00)
2
X(01)
PB2
CM1
SJ300-2 inverter control/ operation function
SJ700

Application Exapmle [4] - Sequential control (holding action, time delay action)

Preliminary

Reference

<Operation example>



1. If PB1 is turned ON, RY1 keeps self holding and turn on T1 and FW command to inverter.

2. If PB2 is turned ON, RY2 keeps self holding and turn on T2 and turn off RY1 and FW command to inverter. T1 off delay provides interlock Interlock in time is provided by T1 off delay.

3. After T1 is turned off, RV command becomes ON.

4. If PB1 is turned ON, RY1 keeps self holding and turn on T1 and turn off RY2 and RV command to inverter. T2 off delay provides interlock.

5. After T2 is turned off, FW command becomes ON.