Industrial rugged Controllers/ Computer ← PLC (Programmable Logic Controllers) → Hardware

Basics -

Can be programmed, reprogrammed

Control machines like assembly line, etc.

written in IEC 61131-3 languages (PLCopen) such as

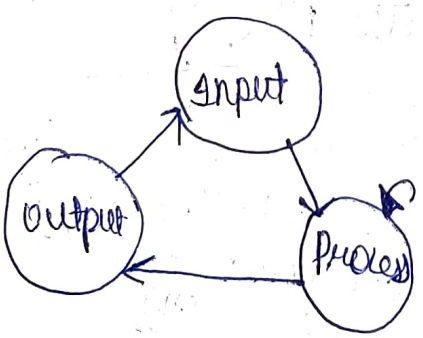
- Structured Text (Letec)
- Ladder Diagram (MOST used) ★
- Functional Block Diagram
- Instruction List
- Sequential Function Chart

DYPIDSE

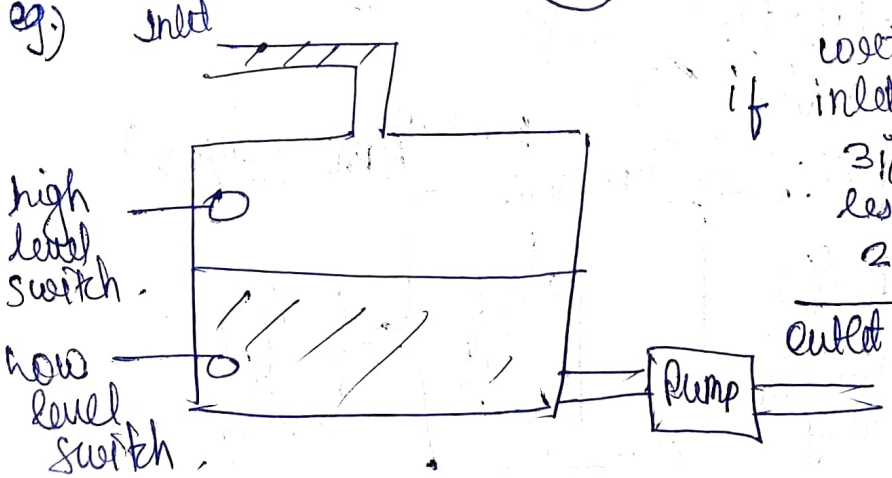
- PLC vs embedded controllers.
  - → rugged
  - → certification

eg (Arduino, Raspberry Pi)

- Need of PLC's
  - Before there were, hard wired logic (electrical components)
    - older control logic
      - → had gates — Binary logic
      - → messy connections
      - (lengthy time, difficult troubleshooting)



eg.)



cost PLC

if inlet is controllable

3 i/p 1 o/p

else

2 i/p 1 o/p

for PLC

DVP - 12SE → PLC type

Controller ~~PLC~~ type Model → 12 – Inputs (8) + outputs (4) (8+4).

12 IO's.

Programmable via Ethernet

grouped in octal format.

Each i/p is binary digital. discrete (0 or 1)

- Powered by 240 AC, 24 VDC (Powered by SMPS)
- Switches, Push Buttons, proximity sensors. (spring)

X0 X1 X2 X3 X4 X5 X6 X7 → 8 i/p - octal coupled.

if had more i/p: X10 X11 ⋮

Discrete digital outputs.

eg LED lamps, relays can be controlled

Y0 Y1 Y2 Y3 → 4 o/p.

2P 4P

→ Switch Mode Power Supply.

100 – 240V AC 1.5A i/p (50-60Hz)

24V DC 2.5A o/p operates PLC

24 V for convenience

+ + – – SMPS L N

line 240 AC neutral

various devices can be tapped. (eg 3 LEDs, 2 push buttons, etc) from 1 SMPS

NPN, PNP wiring

sinking type. common voltage (0V) 24/0

sourcing type.

PNP (24V signal will be received from sensor)

+ 24VDC

24V PLC 0V Data i/p

sensor sinking PNP. sinking

0VDC common

NPN (common) (24/0 V).

+24 VDC

24V PLC i/p

24V sourcing NPN.

0Vdc

S/S Pin — source/sink select Pin for that sensor input (switch b/w 0-24V)

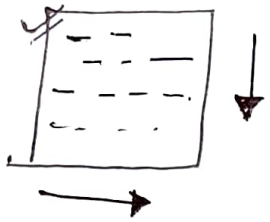
(either use only NPN / only PNP sensors) for 1 PLC

S/S
X0
X1
:

Programming Software (Middle Man)

i/p → Program → o/p.

Scan Time of PLC



Top to Bottom
Left to Right

Scanning from top to bottom
left to right and
returning to to — scan time.

∝ no of i/p, o/p, program order ↑
Scan time ↑

Some sensors give analog o/p
So we need to use extended
analog card module on PLC.

Software
COM manages
Isp Port

PLC selection — controller type

PLC - A series — High end
- redundant CPU, IOs.
if one CPU goes down immediately replaced.

- Compact Modular, Mid range
  A series
  - High end IO > 512, Limited Motion
- DVP Motion CPU.
  Series
  - Motion Control
  - 50MC, 15MC.
  - Ecamp, gearing, interpolation
    (linear, circular (90°))
    (generally)
  - Can have about 60-64 servos.

- DVP series
  - i/p 240V
  - o/p 24V DC — SMPS.
- DVP series slim. CPU.
  - 4 servos 512 i/o points
  - Basic application.
  - Pulse Output
    PWM (Servo)
    will use PTO
    Pulse Train Output
    at very high frequency

COMMGR.
- Add device
- Communicate with PLC

Delta ISP
Programm here
New Project, New File

Ladder Diagram → Download to PLC (Computer → PLC)
Load to PLC
- Upload from PLC (PLC → computer)

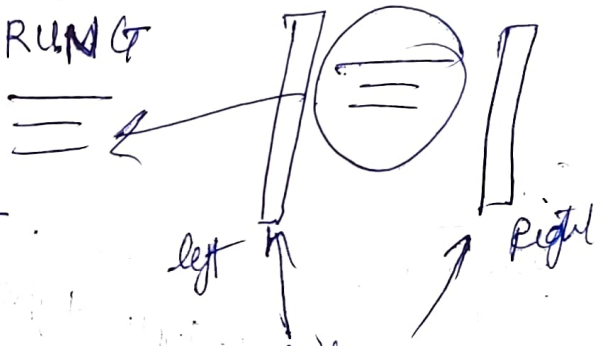
PLC Mode
↳ RUN
↳ STOP

- online (Live with PLC)
↳ (Scan time, steps, RUN available)

Ladder Programming: Graphical Language.

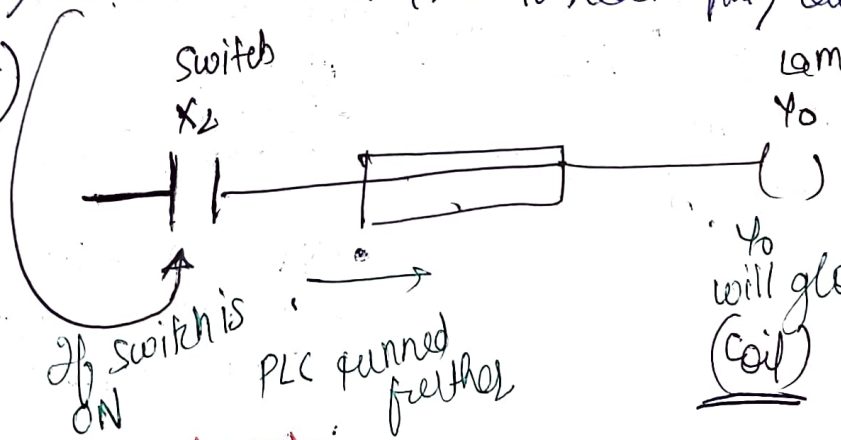
Each line of Programming - RUNG

Logic flows from left → Right.

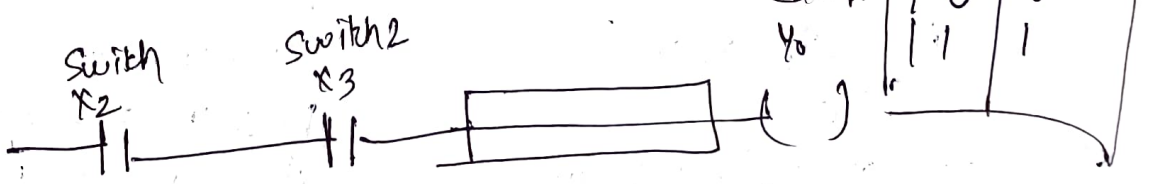


1.) Contact Element (used to read input, output / status)

i)



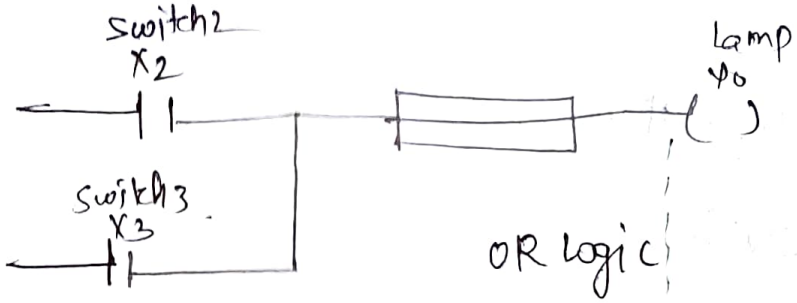
ii)



X2	X3	Y0
0	0	0
0	1	0
1	0	0
1	1	1

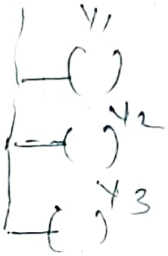
AND logic

iii)



X_2	X_3	Y_0
0	0	0
0	1	1
1	0	1
1	1	1

iv) ~~If all~~ output can also be in series / parallel

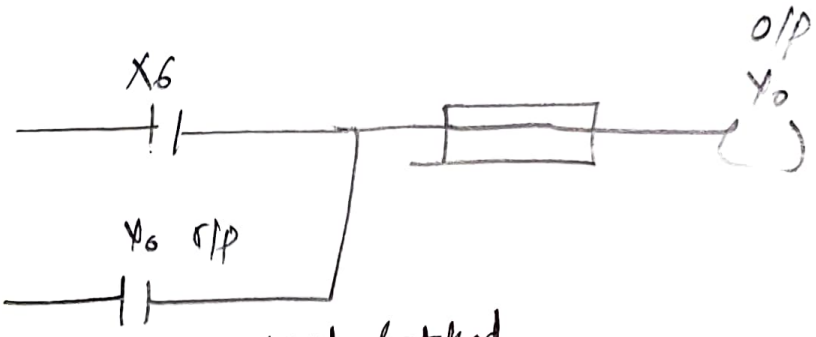


Latching. (Saving state)

eg. Push release Button — Boswell

- Until we press that Button — HIGH state
- It goes off as soon as we release finger.

Hence LATCH required



output latched.

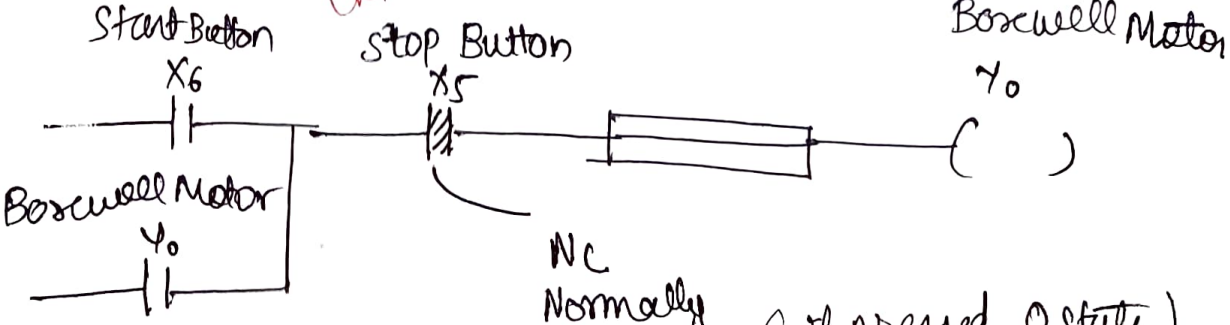
If switch closed / HIGH ckt ON — By default ckt open — NO - Normally open

for switch/pin

NC - Normally closed

By default ckt closed

If switch pressed ckt OFF



SAFETY.

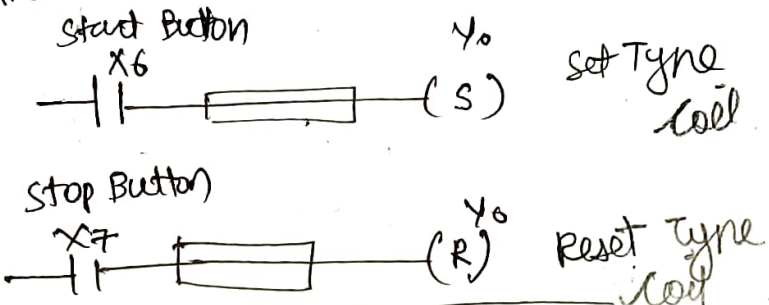
(If pressed 0 state latched)

Y_0 = low.

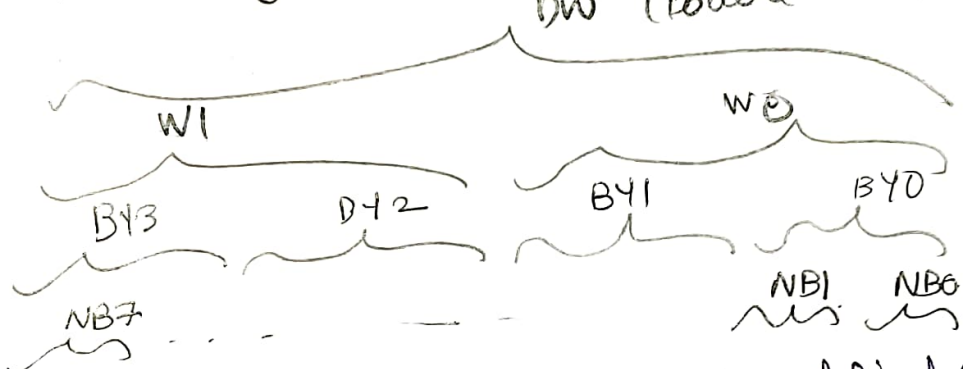
Set and Reset (Different from latching command)
(NO) start & stop button

↑ a concept

Alternative to Boolewell



Memory map in PLC's.



Word
Double (2 word)
Word (2 Byte)
Byte (2 Nibble)
Nibble (4 bits)
Bit (0/1).

Bit (X0, Y0, M0)

↑ Auxiliary Relays / Memory Bit

(can't write to input, so writes to auxiliary relays.)

(No hardware relation)

word (D0)

Timer

- used to introduce time/delay.

double click on ladder

- write eg ld X1.

	Timer	Time Base
T		100ms
		10ms
		1ms.

TMR (S1) (S2)

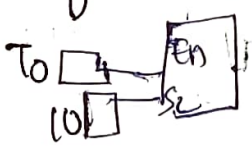
S1 ↑ Time Base

S2 ↑ Time

eg 555ms

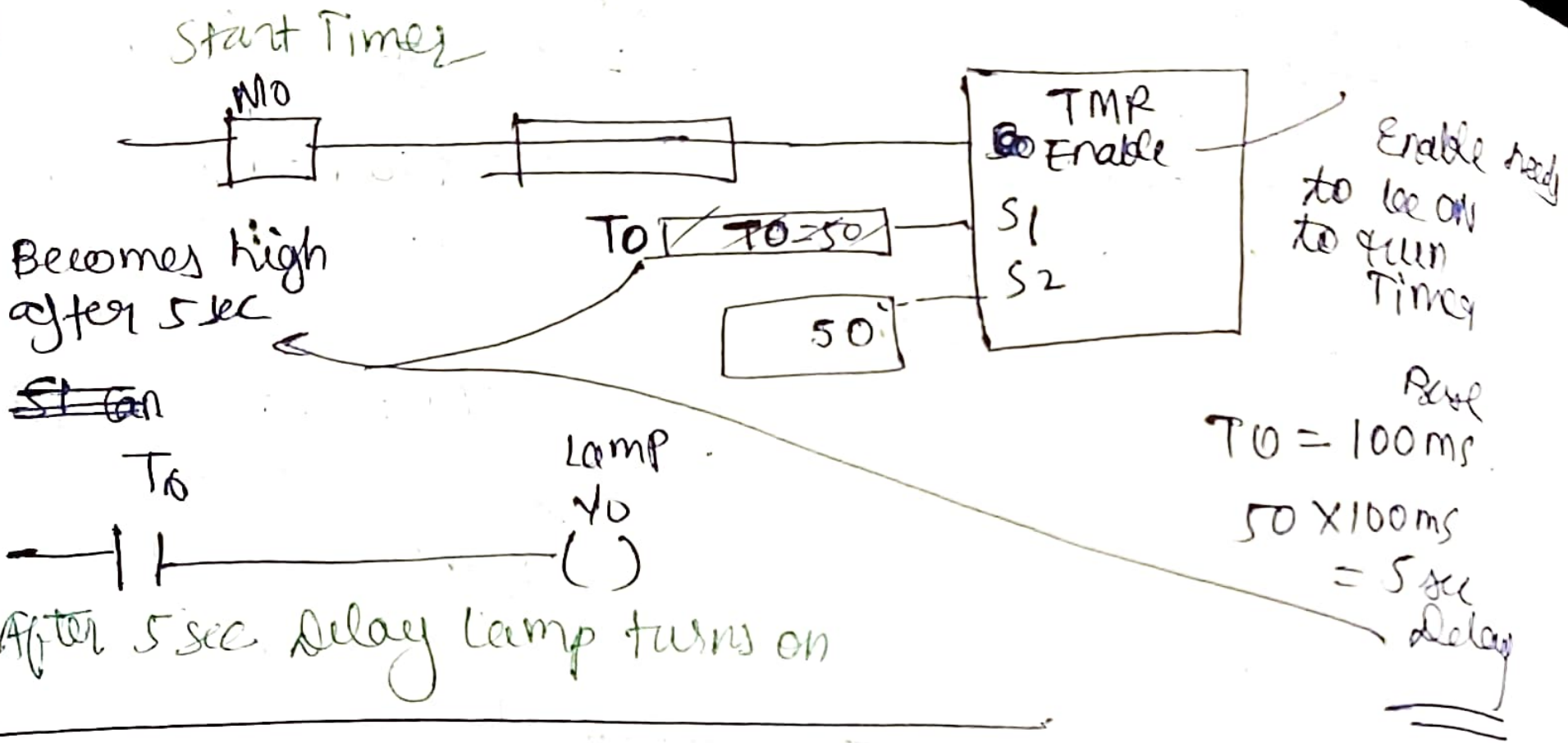
= (100ms) x5 + (10ms) x5 + (1ms) x5

eg TMR T0 10



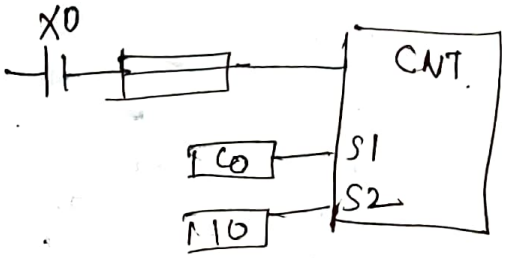
100ms Time Base

eg)


Start Timer
M0
TMR
Enable
S1
S2
T0
T0=50
50
Becomes High after 5 sec
Enable ready to be on to turn Timer
Reset
T0 = 100ms
50 X 100ms
= 5 sec
Delay
T0
Lamp
Y0
After 5 sec Delay lamp turns on

Revise.

Counting :



CNT (S1) (S2)

C0, C1, C2, ...

No of counts

65536

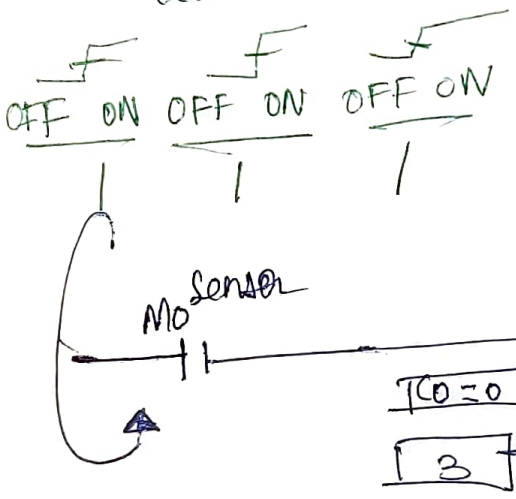
16 bit counter - upto 2^{15}

32 bit counter - up & down

uses encoder (measure position)

Rising edge pulse

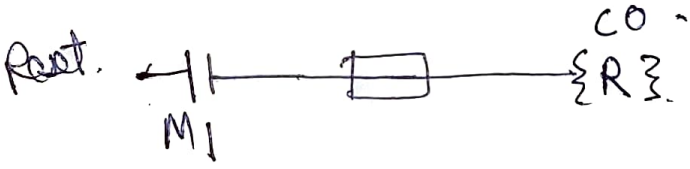
count + = 1.



PLC general

≥ 18V HIGH

< 18V LOW



DOPSoft Application HMI

DOP100

(107WV)

Ethernet

7 inch HMI

Com ports

RS232-485

HMI Human Machine Interface

- Recipe (Parameters / Ingredients) Changeable.

Eg. 5Rs Lays, 10Rs Lays, etc.

→ To interface with HMI – Buttons

Momentary, Maintained

Momentary: like Push Button — until pressed ON, released OFF.

Maintained: One pressed ON/OFF (Switch)

{Data logging part of HMI}

SCADA: ——— (DAview software)

(Supervisory control & Data Acquisition)

Used to control plant

SCADA – software (Can run anywhere, IPC usually Industrial PC)

HMI can be part of SCADA

(MODBUS TCP IP)

raspberry Pi = HMI

MODBUS Serial

EtherCAT, CanOpen } servos

DIAView – Scada Software

- Drivers for different company hardware. Report generation

↳ Variable dictionary : Master Variable

Scada has tags – No of variable

(Add like that)

M0, M1, ... → Auxiliary Relay – (Digital)

D0 – Word (Analog)

Servos & Drives:

– High speed, High Accuracy Rotation

(Any Rotation motion ⇒ Linear motion)

↳ Ball Screw Mechanism

– AC 3 phase Induction Motors (Runs on 1/3 phase AC)

↳ when power supplied (240V) it will always run at ~~240~~ its rated RPM

To control speed ~~(servo)~~

Use a VFD with AC motor (like a Drive)

↳ Variable Freq. Device

⇒ Changing power frequency → Different speed control

PLC → VFD → Motor

(Eg conveyor) ✱✱✱ [VFD + AC Motor] → eg Cranes (Req position also)

VFD – Open loop – No position control.

✱ Not as accurate like stepper/servo drive ← ENCODER: GIVES POSITION FEEDBACK (Closed Loop)

SERVO: (Closed loop control) Robotics

Steppers and Servos

Steppers → (Hobby Projects) → Open Loop → Feedback to drive → For >1000 rpm torque (force) gets reducing (Can bear less load)

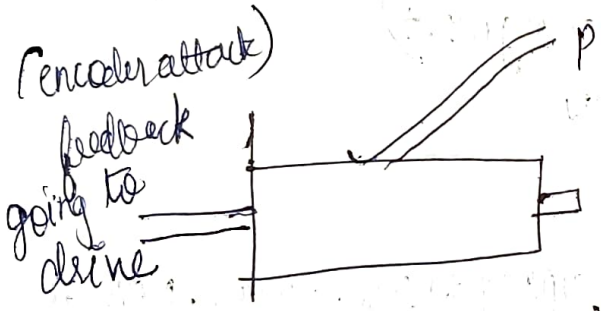
Servos → Closed Loop → Feedback to drive → Servo provide HIGH speed & HIGH torque → Also work at >3000 rpm.

Types of servo

1. AC / DC — High torque density
2. Brushes / Brushless
3. Synchronous or Asynchronous

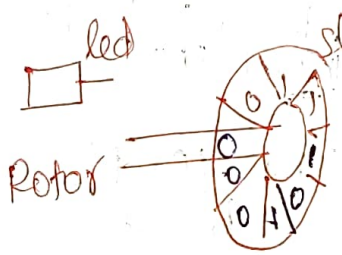
AC Servo - Easy

AC Brushless - Most Synchronous used.



(Encoder is sensitive)

* We can also take feedback in PLC - Our choice
  1. Encoder Feedback ~~ego~~ (Asian) ✗
  2. Resolver Feedback. (European)



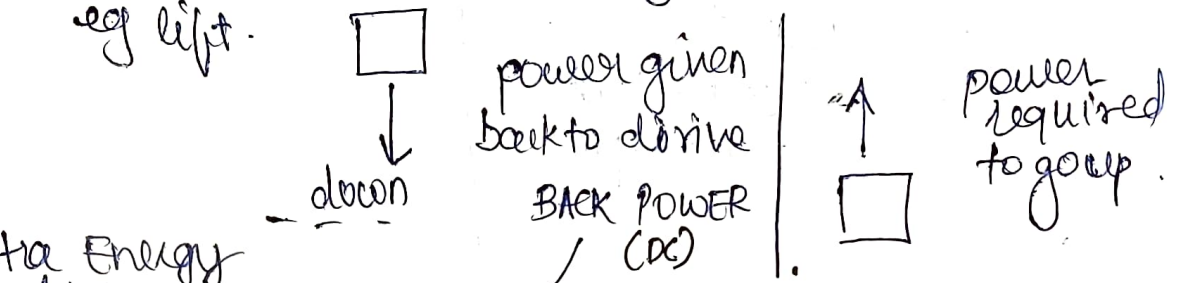
001 — 45°
001 — 90°
000 — 135°
100 — 180°
010 — 225°
101 — 270°
110 — 315°
111 — 360°

Servo
Nominal Torque
~~2000~~ Nm 1-27
~~= 3000~~ rpm

3000 rpm

ECMA - C 1 0 60 Z E S .

Motors: can work as motoring as well as generator
eg lift.



Can use BACK POWER again (in case of two lifts)
done using DC Bus.

Servo ≡ Controller
↳ display on servo drive (7 segment)
↳ has over 2000 parameters
↳ we set 3-4
↳ Based on Modes ≡ can display torque, load, RPM etc
↳ setting parameter via display, Buttons
↳ standard used Modes.
1. Position control Mode → External position mode controlled using pulses from PLC
→ Internal position mode

2. Speed controlled → External PLL
→ Int

3. Torque controlled → External PLC
→ Internal

Internal mode: can control servo just by servo drive also.

can give i/p using digital cable (bits)

Generally, external PLC/controller used.

6-7 external I/Os

Position Control.

→ Position command (move conveyor to particular position) from position a to b.

Speed control / velocity

→ we control this using 0-10V Analog i/p / potentiometer

rotate at x rpm.

Control signal.

0-10V ≡ 0-3000 rpm

1V ≅ 300 rpm

0.001V ≅ 3 rpm

Torque Mode ★

Different for different Bottle caps.

X Position

X Torque

Programming

(Y0 Y1 — generate pulses) — connected via a interface cable

eg Y0

Pulse train output

Duty cycle of PTO = 50%

Same

★

no of pulses generated ≡ POSITION

(PPR: Pulses / Revolution) defined in servo

for 1 Rot of servo — How many pulses will get from feedback?

How many pulses to give from PLC

eg 10,000 PPR

1 Rotation → 10,000 pulses

High PPR ; High Accuracy

$\frac{360^\circ}{10,000}$

0.036°/pulse

as PPR ↑ amount of pulses generatable per sec ↑

in Siemens PLC – max freq = 100KHz = f_{max}

∴ 100,000 pulses/sec · PPR = max possible

eg. 1 Rotation = 10,000 pulses. (servo data) for set FREQUENCY → Speed

1 Rotation/sec.

⇒ 60 Rotations/min → SET POSITION

freq = 10,000

per sec → 10,000 pulses

10,000 pulses per rotation in 1 sec.

1 sec 1 rotation

∴ In 1 min – 60 times rotation

60 rpm

≡ 3000 rpm ⇒ 50 × 10,000 = 5 lakh pulses

as programmer we used pulses

ISPsoft

DPLSY S1 S2 D

S1 → Pulse O/P frequency

S2 → No of pulses

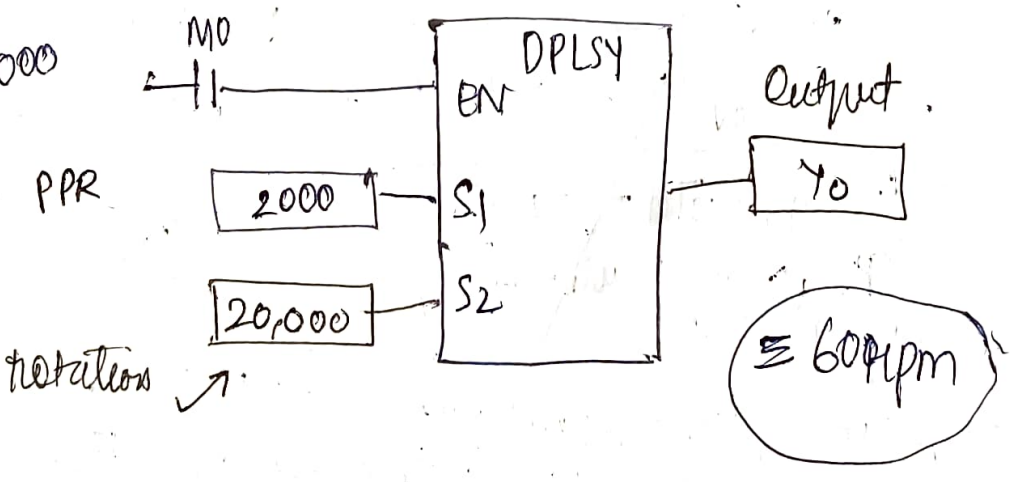
D → Ext O/P.

eg. DPLSY 10000 250

eg

PPR = 2000

≡ 60 rpm



I want to reach 500 rpm max
100kHz ✓

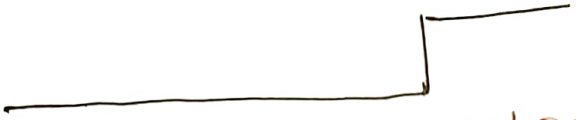
PPR = 2000
↳ 60 rpm

360 ÷ 2000 = 0.18
(Low accuracy best High speed)

PTO servo control

Y0

Y1 (direction)



Either LOW = FORWARD / REVERSE
HIGH = REVERSE / FORWARD

If servo is OFF can change position of shaft.
servo is ON cannot change position.

Position doesn't change, after x revolutions even

→ DPSLR → Acceleration, Deceleration

Motion controllers
CANOpen
Ethercat