



MAIN

MP3 PLAYERS

CUSTOM EL LIGHTING

LINKS

DODGE NEON



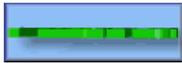
PROJECTS

KITS

Misc. Pictures

MUSIC

FORD F250 4X4



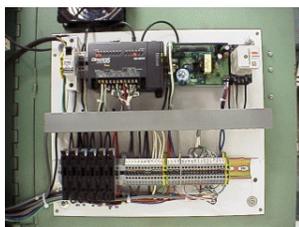
ELECTRONIC STUFF

MOTORCYCLE

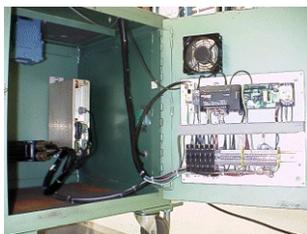


This puller is driven by a Pacific Scientific servo motor and 930 drive.

Automation was accomplished by using a [Automation direct](#) DL05 PLC. The operator interface is a Optimization panel connected to port 1 on the PLC. The Servo system communicates with the PLC via Port2.



This is the panel in the servo puller. This included a DL05 P.L.C. and 24 Volt D.C. power supply by Automation Direct



This shows the orientation of the panel and motor, drive systems.

The project started out as a way to remove the Windows / Dos based PC that was used to run the system. Using the PC was slow and bulky, since this puller was a “portable” unit used through out the plant, having a PC to move around with it was a pain. And using the Pacsi software to run the drive was also a pain, plus it was based in motor RPM’s. We needed engineered units equal to feet per minute as the base to work with.

Using the DL05 as the brains of the unit. I added an Optimisation OP1500 to display running speed, pulled distance and allow operator to change speed. Speed and distance was acquired by using a 600 Pulse per revolution encoder using a 12” diameter wheel making contact with the pulling belts. The encoder was connected to the PLC via inputs X0 through X1 and the PLC was configured using Mode 20, high-speed counter input and Quadrature count mode.

Run, Stop and Speed adjust keys were assigned to the OP1500’s 5 programmable keys. The operator can key in the needed process speed by pushing the speed key and then typing in the required speed. To make fine speed adjustments I put a digital pot on the machine. The pot is a 12 pulse per revolution digital counting encoder with a ¼ inch shaft that a standard knob can be mounted on. When the operator turns this pot, a digital A and B signal is sent to the process via inputs X4 and X5. The ladder logic then decodes whether A lead B or if B lead A. This tells the machine whether to count up or down. The count is then added or subtracted from the requested speed.

The tricky part was the communication with the servo drive. The OC930 card receives a serial command that is ASCII based. It includes the servo address, command, command value and a checksum. The command value and checksum are the only parts that would change. What made it even more difficult was that the command value was a hexadecimal representation of an IEEE 754 floating-point number. (Also known as a REAL number in the Automation Direct PLC’s). Since the DL05 does not support REAL numbers like some of the DL205 PLC’s, (it has no “BTOR” binary to real or “RTOB” Real to binary command, I had to create this number using ladder logic. I will not go into great detail about the ladder logic, suffice it to say that the conversion and the modulo 256 checksum was 75 percent of the ladder program. This command string, once assembled, would be printed out the second comm. port and sent to the servo drive.

Last Updated on 4/12/2004

By Len Averyt

Email: Techknowman@techknowman.com