Clock Synchronization Protocol:

Port 0 goes from the computer to the robot

Port 1 goes from the robot to the computer

IO line on port 0 corresponding to segment:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NI USB-6525 DAQ** | |  | **Fanuc** | **Protocol Info (transmitting digits)** | **Protocol Info (transmitting seconds)** |
| **Port** | **Line** | **Direction** | **I/O Register** |  |  |
| 0 | 0 | → | DI [101] | Computer Sets High when information is ready to be read | Computer Sets High when information is ready to be read |
| 0 | 1 | → | DI [102] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 2 | → | DI [103] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 3 | → | DI [104] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 4 | → | DI [105] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 5 | → | DI [106] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 6 | → | DI [107] | 1 when segment is lit, 0 otherwise (see figure above) | Binary Seconds |
| 0 | 7 | → | DI [108] | 1 when segment is lit, 0 otherwise (see figure above) | 0 if AM, 1 if PM |
| 1 | 0 | ← | DO [105] | Robot sets to high when it wants any information. When it goes low, P0-0 should also go low |  |
| 1 | 1 | ← | DO [106] | Robot sets high when it wants 1st digit |  |
| 1 | 2 | ← | DO [107] | Robot sets high when it wants 2nd digit |  |
| 1 | 3 | ← | DO [108] | Robot sets high when it wants 3rd digit |  |
| 1 | 4 | ← | DO [109] | Robot sets high when it wants 4th digit |  |
| 1 | 5 | ← | DO [110] | Robot sets high when it wants Seconds or AM/PM |  |
| 1 | 6 | ← | DO [111] | Not Used |  |
| 1 | 7 | ← | DO [112] | Not Used |  |

When P1\_7 is low, program starts.

Sets pins 0 through 6 for first digit, digit on right.

Sets pins p0\_7 high, safe to read.

Stays at digit while P1\_7 is low.

When P1\_7 goes high, P0\_7 goes low.

Read P1\_7. When goes low, send next digit.

1,2,3,4,seconds, repeat