

## Board-Level Programming for the EDE1200 Robot

The 595 shift register provides 8 bits (one byte) of memory. The memory is set by serial signals from the calculator. The 8 memory locations (A through H) are then available as a parallel output to hold the EDE1200 controls in any desired configuration. The 595 shift register also has a serial data output line, which allows the user to “daisy chain” two or more shift registers and therefore control 2 or more boards from a single calculator.

Data must be sent in the correct order. Each data bit pushes (or “shifts”) all the previous data bits over one place. If all 8 bits are full, the data is pushed on to the next 595 in the daisy chain. The write command must therefore start with the “H” value for the board furthest from the interface, work its way up to the “A” value for that board, and then repeat the process for the board closer to the calculator. The CBL2 and LabPro provide a “2001” command that is designed to make this process easier. The process with the original CBL is somewhat more cumbersome, although the CBL has the advantage of an automatic “clock” signal, something not available on the CBL2 or LabPro.

Data signals to the 595 shift register must be either “1” or “0” (either +5 volts or 0 volts), and these signals are provided by the D0 ( $2^0 = 1$ ) digital output line of the CBL, CBL2 or LabPro. On the original CBL, this is the red wire from the DIG OUT port. On the CBL2, this is the blue wire from the DIG/SONIC port. The LabPro has 2 DIG/SONIC ports, and DIG/SONIC 1 functions precisely as described for the CBL2.<sup>1</sup>

The shift register must be “clocked”—told that data is ready—before it will accept data from calculator. This clock signal is provided automatically by the CBL over the white wire, so all the CBL needs to send is a sequence of data values. With the CBL2 and LabPro, we use the D2 ( $2^2 = 4$ ) line for the clock signal. To transmit the data bit “1,” we must send the sequence, “1,5”. To send the data bit “0”, we must send “0,4”.

The process above transmits data to the shift register, but it does not yet activate the shift register’s parallel output lines. If the output happened immediately, the EDE1200 chip would try to execute each instruction before the entire set is in place. The “latch” is the signal to the shift register that all the commands are in place and ready for execution. We use the D1 ( $2^1 = 2$ ) line (green on the original CBL, yellow on the CBL2 and LabPro) as the latch. The signal for the shift register to execute a set of commands is therefore “2,0” where the trailing “0” simply returns the interface outputs to zero volts.

The “2001” command on the CBL2 and LabPro is limited to 32 elements, but it takes 34 elements to setup and “latch” 2 shift registers. The elements must therefore be sent in two parts, and we generally sent the final latch command, “2,0”, separately.

---

<sup>1</sup> The instructions here assume that LabPro users have connected the 595 Shift register to DIG/SONIC 1. The same functionality can be achieved with a shift register connected to DIG/SONIC 2, if all values A through H are increased by adding 256. The DIG/SONIC 2 port does not function correctly under early versions of the LabPro firmware. Before attempting to use DIG/SONIC 2 for outputs, be sure you have upgraded the LabPro firmware to version 6.23 or later. The upgrade is available free at [www.vernier.com](http://www.vernier.com).

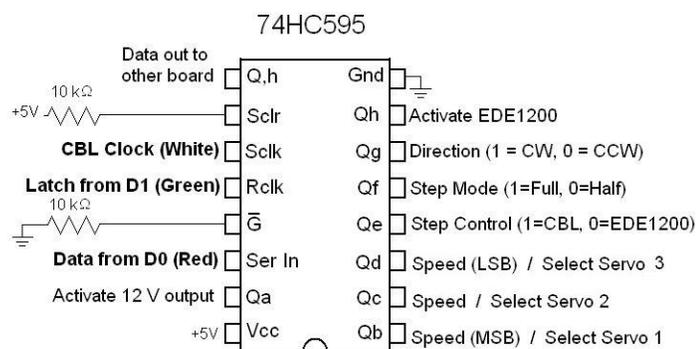
FROM THE ORIGINAL CBLTo set and latch the 8 output lines on ONE board

```
Send({1,31,10,H1,G1,F1,E1,D1,C1,B1,A1,2,0})
Send({3,.0002,10,0,0,1,0,0,0,0})
```

To set and latch the 16 output lines on TWO boards

```
Send({1,31,18,H2,G2,F2,E2,D2,C2,B2,A2,H1,G1,F1,E1,D1,C1,B1,A1,2,0})
Send({3,.0002,18,0,0,1,0,0,0,0})
```

## Shift Register Pin Connections for the Original CBL



FROM THE CBL2 or LabProTo set and latch the 8 output lines on ONE board

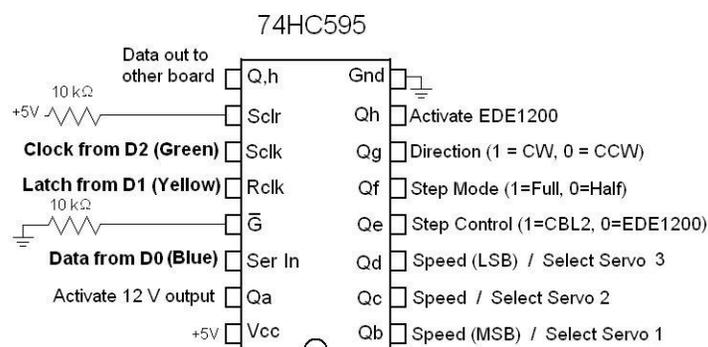
```
Send({2001,H1,H1+4,G1,G1+4,F1,F1+4,E1,E1+4,D1,D1+4,C1,C1+4,B1,B1+4,A1,A1+4,2,0})
```

To set and latch the 16 output lines on TWO boards

```
Send({2001,H2,H2+4,G2,G2+4,F2,F2+4,E2,E2+4,D2,D2+4,C2,C2+4,B2,B2+4,A2,A2+4,H1,H1+4,G1,G1+4,F1,F1+4,E1,E1+4,D1,D1+4,C1,C1+4,B1,B1+4,A1,A1+4})
```

```
Send({2001,2,0})
```

## Shift Register Pin Connections for the CBL2 or LabPro



**Shift Register Outputs:**

Send in order	595 Output	EDE1200 Pin	Function
1	H	6	Activate: 1 = do it, 0 = free spin
2	G	7	Motor Direction: 1 = clockwise, 0 = counterclockwise
3	F	8	1 = Full step, 0 = Half step
4	E	10	Clock select: 1 =CBL step, 0 = EDE1200 clock
5	D	11	Speed LSB <b>or</b> Select Servo 3
6	C	12	Speed <b>or</b> Select Servo 2
7	B	13	Speed MSB <b>or</b> Select Servo 1
8	A	NA	Laser switch on = 1, off = 0

**Controlling the Servo Motors:**

The D3 ( $2^3 = 8$ ) line is not used in the procedures which control the stepper motors and the 12-volt output. This line is used instead to control servo motors, but there is also a need to address the signals from the D3 line to a particular servo. This is done by using the 595's "speed" lines as address line. The output signal to servo motors is the result of using an "AND" gate to combine the D3 signal with the 595's outputs, D, C and B. The assumption is that servos will only be moved while the robot is not moving, so that the speed controls are then free to serve this second function. It is also possible, however, to move the servos while the robot is moving under control of the EDE1200 chip—provided the speed choice is coordinated with the choice of servo address.

Moving a servo requires that the correct 595 output first be turned on, using the commands above. At that point, a correctly timed series of pulses on the D3 line will cause the servo to move without changing the other servos.

For example, the first two lines in the sequences shown below will zero all 595 outputs on two boards, except the output which enables servo 2 on the board nearest the calculator. The final two lines move the servo to a position that is determined by the pulse width, P.

**FROM THE ORIGINAL CBL**

```
Send({1,31,18, 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,2,0})
Send({3,.0002,18,0,0,1,0,0,0,0})
Send({1,31,16,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0})
Send({3,P,512,0,0,1,0,0,0,0})
```

**FROM THE CBL2 or LabPro**

```
Send({2001,0,4,0,4,0,4,0,4,0,4,0,4,0,4,0,4,0,4,0,4,1,5,0,4})
Send({2001,2,0})
Send({1,31,16,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0})
Send({3,P,1600,0,0,1,0,0,0,0})
```