SD20 - 20 Channel I2C to Servo Driver Chip

The SD20 chip is a pre-programmed PIC16F872 running at 8MHz. Position commands for up to 20 servo's are sent to the chip over the I2C bus.

Features Include;
Easy I2C bus control, same protocol as popular EEPROM's such as 24C02
Position Control of up to 20 Servo's
Servo's inactive at power up until first position sent to it
Scan time is 20mS minimum, then proportional to the total number of servo's used.
Buffer can be read as well as written saving valuable memory resources on small controllers.
Standard mode - 256 bit resolution from 1mS to 2mS.
Expanded mode - Allows pulse to exceed 1mS to 2mS range for greater movement.
0.5uS Timer produces jitter free outputs.

Connection Diagram showing example connection to OOPic.

Note1. Connect 100n Capacitor between +5v and GND close to CPU pins.

Note2. Connect 4k7 pullup resistor between Servo5 and +5v, as pin 6 (RA4) is open Drain output.

Note3. If possible, use separate supply 4.5v to 7v for servo's

8MHz Ceramic Resonator

SD20 - Top View
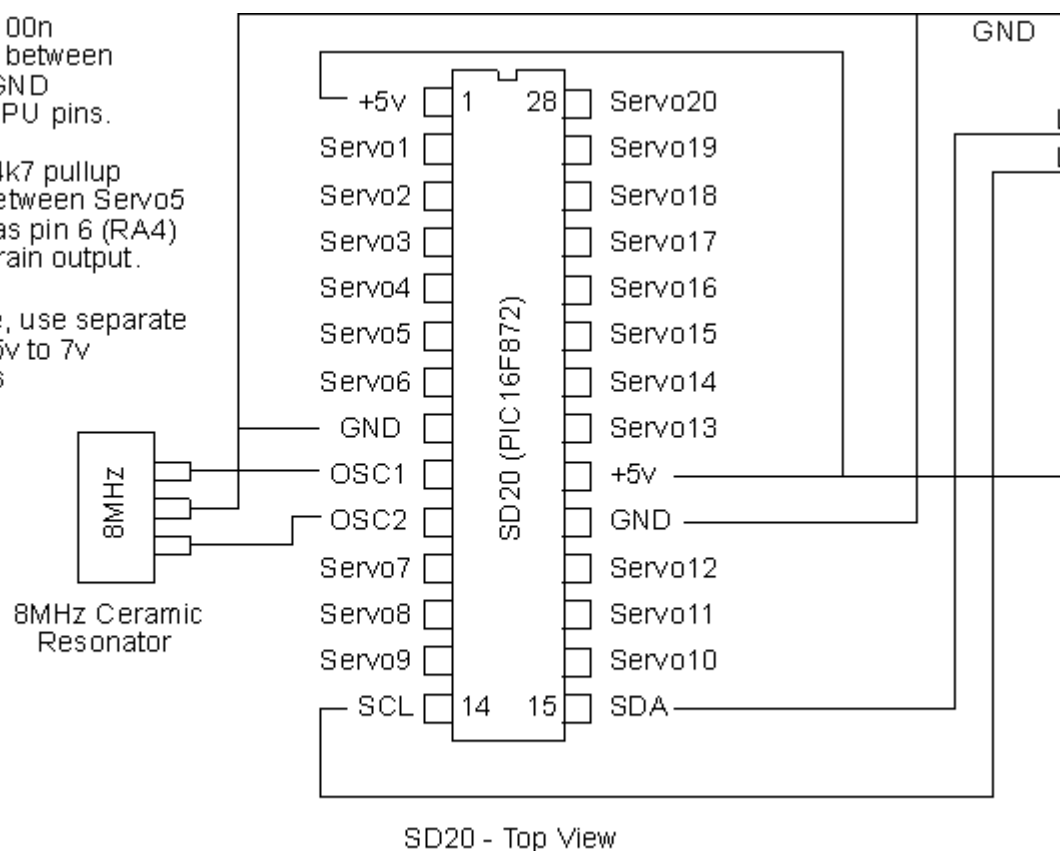
OOPic Conr

Register Usage

Register	Function
0	Software Revision Number
1	Servo1 Position
2	Servo2 Position
3	Servo3 Position
4	Servo4 Position
5	Servo5 Position
6	Servo6 Position
7	Servo7 Position
8	Servo8 Position
9	Servo9 Position
10	Servo10 Position
11	Servo11 Position
12	Servo12 Position
13	Servo13 Position
14	Servo14 Position
15	Servo15 Position
16	Servo16 Position
17	Servo17 Position
18	Servo18 Position
19	Servo19 Position
20	Servo20 Position
21	Standard/Expanded Mode Control
22-23	Expanded mode offset - 16bit register, high byte first
24 - 31	Unused bytes - available to user

Standard mode
This is the power-up state. The servo range is the standard 1mS to 2mS. You can restore this mode, if you have changed it, by writing 0 (zero) to register 21. Writing values from 1 - 255 (0x01 - 0xff) will produce a pulse range as described in Expanded Mode below.

Expanded mode
Be careful with this, as it is easy to cause the servo to hit its stops.
This mode globally allows an expanded range of movement of the servo's. Writing any value greater than zero to register 21 will set Expanded Mode. The actual value written sets the range of the pulse width according to the following formula;
Range Width (uS) = (255 * 256) / Reg21
where **Range Width** is the *Variation* in the width. So for example, if Reg21 is set to 65 the the maximum variation in pulse width will be;
(255*256)/65 = 1004uS = 1.004mS.
This is just the *variation* in range, for an actual range of 1mS to 2mS an offset needs to be added to this. The offset is contained in register 22-23 and is;
(Reg22:23 +20) (uS)
So if Reg22:22 is set to 980 then the offset will be 980+20 = 1000uS = 1mS.
Therefore the pulse will vary from 1mS to 2mS. This is the same as the standard range.
The total pulse width is;
Pulse Width (uS) = ((255 * 256) / Reg21) + (Reg22:23 + 20)
Values of 43 and 820 will give a range of 0.84mS to 2.36mS. This is a good wide range on the HS300 servo.

Scan Time
This is the refresh time between pulses to a servo. No servo is active until its position is set. The refresh period for the servos depends on the number in use and their pulse width. It is maintained at 20mS as far as possible. When the required time exceeds 20mS then it expands as required,

according to the number of servo's used and their pulse widths. Any SD20 channels may be chosen, because unused channels do not add to the scan time. You don't have to start with position 1. On Rev1 devices, writing 0 to a servo will move it to the minimum position, on Rev2 devices, writing 0 will de-activate the servo. This allows a servo to be powered down by removing drive pulses.

Software Revision Number
Register 0 is loaded with the software revision number after a power-up. It can be over-written and the register used as an additional byte if required.

Address
The SD20 Servo chip is located at address 0xC2 on the I2C bus.

Example Code
 The following code demonstrates the SD20 used with the OOPic.

```
Dim SD20 As New oI2C              ' Create the servo objects
Dim val As New oByte


Sub main()
  SD20.Node = 97                  ' Decimal of Hex address 0xC2 shifted right by 1
  SD20.Mode = cv10bit             ' I2C mode uses 8-Bit Register Address
  SD20.NoInc = 1                  ' Don't increment


' The following 6 lines set expanded mode and can be deleted for 1mS-2mS operation
  SD20.Width = cv8bit             ' 8-bit wide data
  SD20.Location = 21
  SD20 = 43                       ' Set Expanded mode
  SD20.Width = cv16bit            ' 16-bit wide data
  SD20.Location = 22
  SD20 = 820                      ' Set offset


  SD20.Width = cv8bit             ' 8-bit wide data
  SD20.Location = 17              ' Use Servo 17 for demonstration
  While(1)
    val = SD20
    val = val + 1                 ' This will move servo slowly forward,
    SD20 = val                    ' with fast reset and repeat
  Wend
End Sub 'main
```