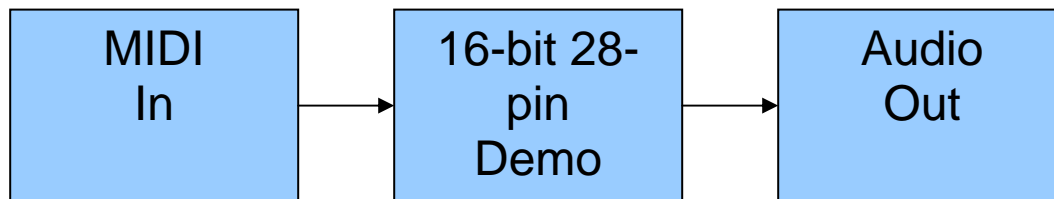


Registration Number: MT2290

Parts Used: dsPIC33FJ12GP202 and MCP4921

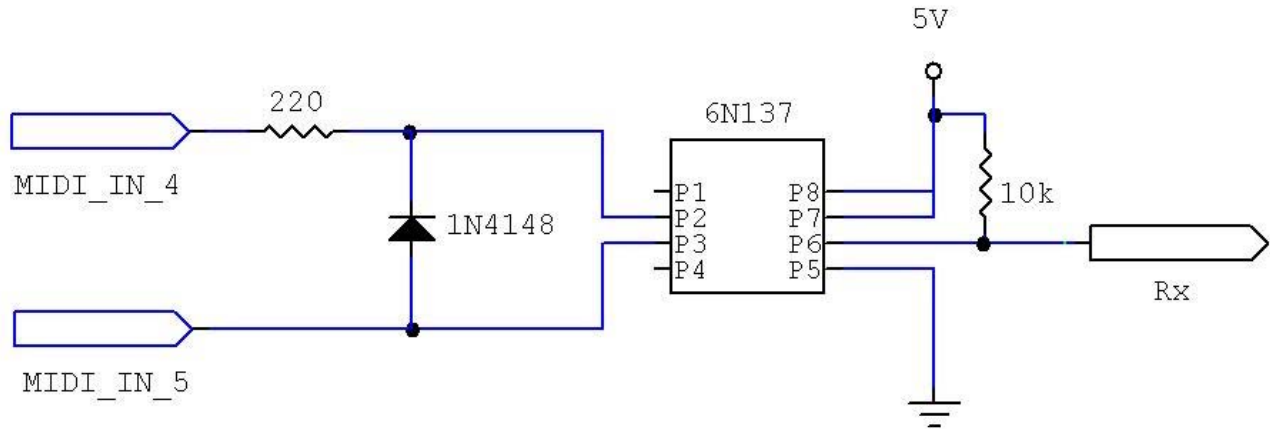
### dsPIC\_Synth Abstract

This project uses a dsPIC33 plugged into a 28-pin demo board with an MCP4921 12bit SPI DAC and a 6N137 optoisolator to create a single voice MIDI synthesizer. The board plugs directly into the MIDI out of a keyboard or sequencer and responds to MIDI “note on” and “note off” commands in real-time with a harpsichord/organ-like sound. To make the sound richer, two overtones are used. The code for the project is written entirely in C demonstrating the efficiency and performance of the dsPIC33 and the Microchip C compiler.

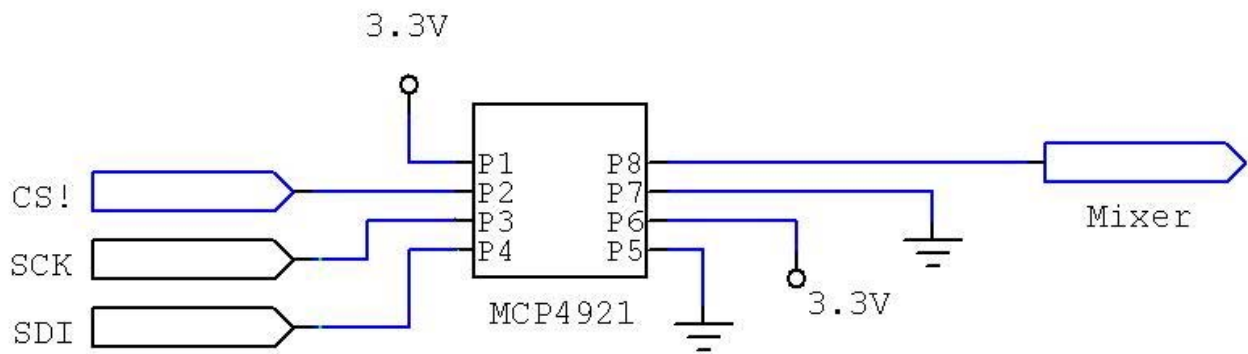


Project Block Diagram

The following was added to the 16-bit 28-pin demo board:

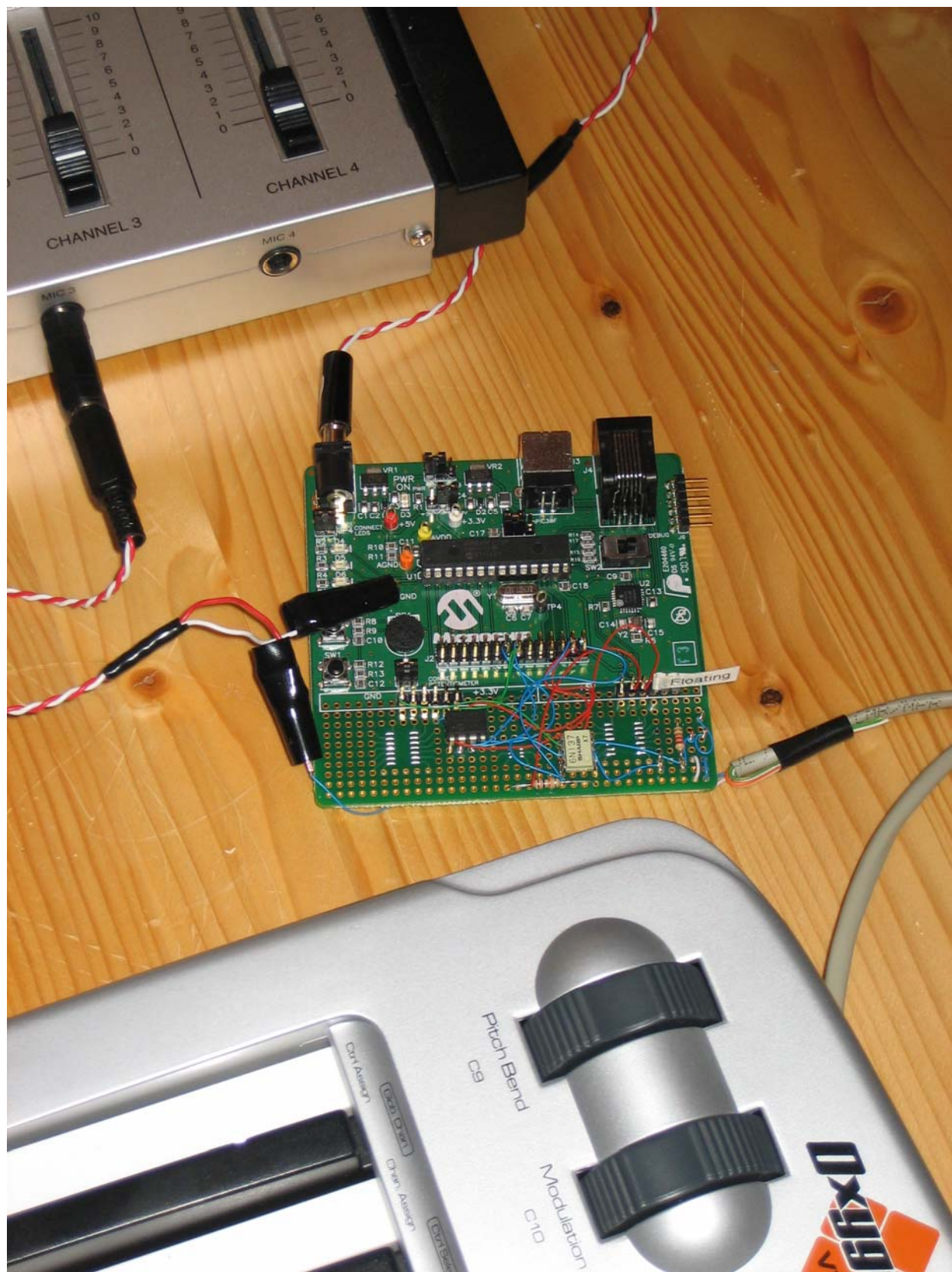


Rx->RP6

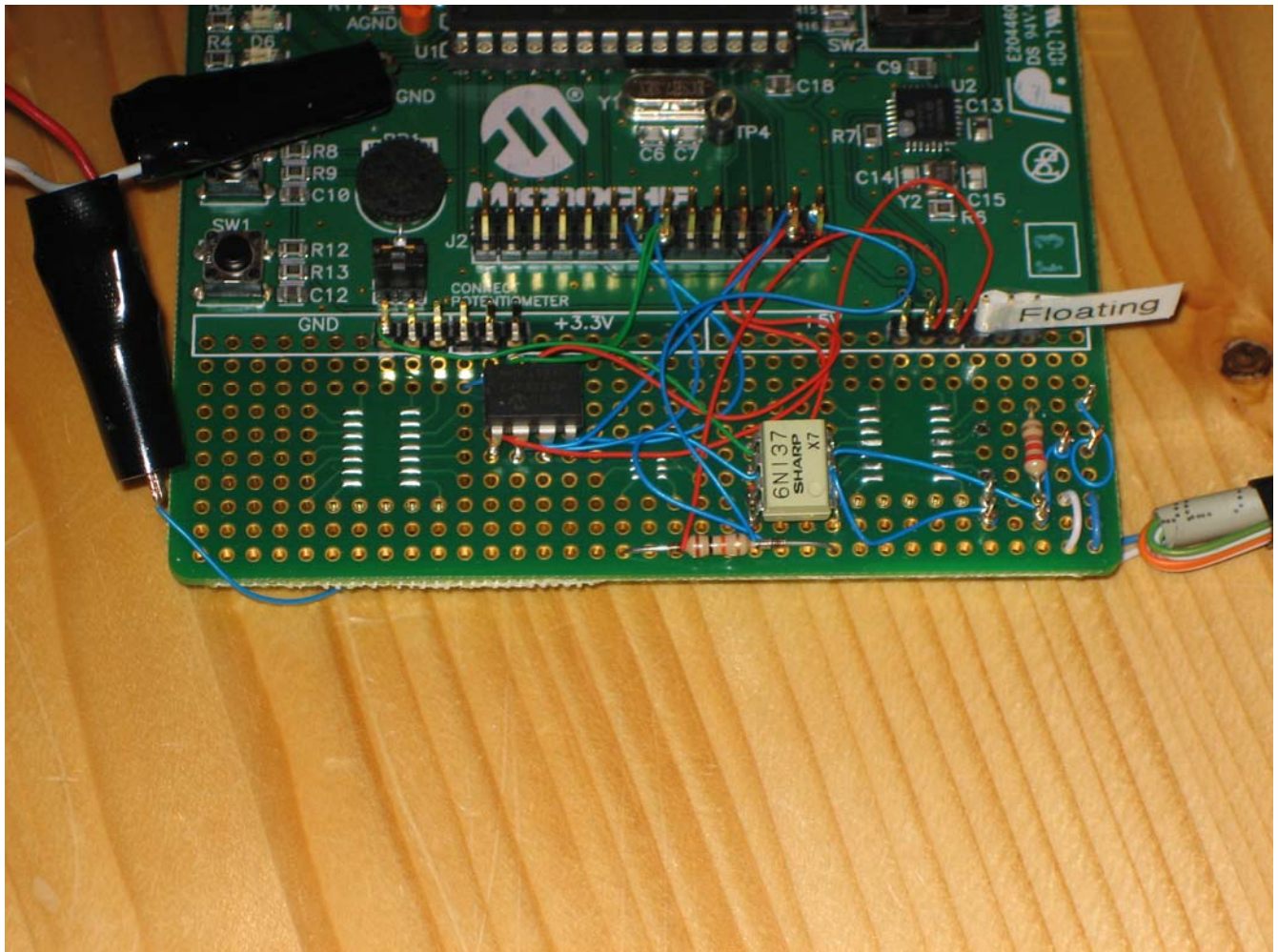


RB7->CS! RP10->SCK RP11->SDI

A picture of the completed assembly including the MIDI keyboard and mixer:



A close up of the prototype area of the demo board:



## A code sample of the timer ISR

```
void __attribute__((interrupt)) _TlInterrupt(void)
{
    int theWave, theWave1, theWave2, theWave3;

    IFS0bits.TlIF = 0;           // clear interrupt flag

    if (sub_second_counter++ == 32767) // Do this every second
    {
        sub_second_counter = 0;
        seconds_counter++;
    }

    if (gWave.state)
    {
        theWave1 = wave_lookup(gWave.angle1);
        theWave2 = wave_lookup(gWave.angle2)/2;
        theWave3 = wave_lookup(gWave.angle3)/4;

        // Scale the wave to [-2047,2047]
        theWave = (theWave1 + theWave2 + theWave3)/16;

        // Bias to 2048
        write_SPI(theWave + 2048);

        if (gWave.state == 3) // note transition
        {
            // Prevent clicking on note-to-note transitions
            // by transitioning near zero
            if (theWave > -32 && theWave < 32)
            {
                gWave.state = 1;
                gWave.freq1 = gWave.next_freq1;
                gWave.angle1 = 0;
                gWave.freq2 = gWave.next_freq2;
                gWave.angle2 = 0;
                gWave.freq3 = gWave.next_freq3;
                gWave.angle3 = 0;
            }
            else
            {
                gWave.angle1 += gWave.freq1;
                gWave.angle2 += gWave.freq2;
                gWave.angle3 += gWave.freq3;
            }
        }
        else if (gWave.state == 2) // note off
        {
            // Prevent clicking on note off
            // by waiting until near zero
            if (theWave > -32 && theWave < 32)
                gWave.state = 0;
            else
            {
                gWave.angle1 += gWave.freq1;
                gWave.angle2 += gWave.freq2;
                gWave.angle3 += gWave.freq3;
            }
        }
    }
}
```

```
}
else // if (gWave.state == 1) // continue playing
{
    gWave.angle1 += gWave.freq1;
    gWave.angle2 += gWave.freq2;
    gWave.angle3 += gWave.freq3;
}
}
else
write_SPI(2048); // Quite to bias to prevent clicking
}
```