Column #59, March 2000 by Lon Glazner:

A Digital-to-Analog Converter for All Seasons

The digital-to-analog converter (DAC) is relatively common in embedded control designs today. Many circuits respond best to analog inputs, which leads to a requirement for digital front ends. These digital front ends then interface to various controlling entities, or sensors, or both.

The BASIC Stamp is an ideal device for controlling a DAC. These DAC and BASIC Stamp working in conjunction, and connected to other simple circuitry, can meet a large variety of analog interface requirements. I'll cover a powerful and simple circuit that is flexible enough for a myriad of designs which might require analog control signals.

Defining the Design

So I'm sitting behind the desk one day, playing Lunar Lander on my personal digital assistant and, between fiery crashes, the Grand Pooh-bah approaches me with a design requirement. He's got a customer that needs a controller for some custom lighting. This lighting system requires analog input voltages ranging from 1-24VDC, one control signal per lamp. The load on this analog control voltage is expected to be light, no more than 10-20mA. The end system will be part of an RS-485 network but, for the short term, the device needs to be a single control node. The RS-485 and communication/control protocol will be tackled later in the design cycle. One last got-cha, the whole thing has got to fit in a 3"x2" box and each node must control four lamps.

After the Grand Pooh-bah shambles away, I'm faced with a daunting task. How can I quickly get back to my Lunar Landing practice? After all, NASA could call at any time looking for that one engineer capable of rescuing stranded astronauts from our rocky neighbor in the sky. This design calls for a BASIC Stamp and, in particular, the BASIC Stamp 2 (BS2).

The Hardware

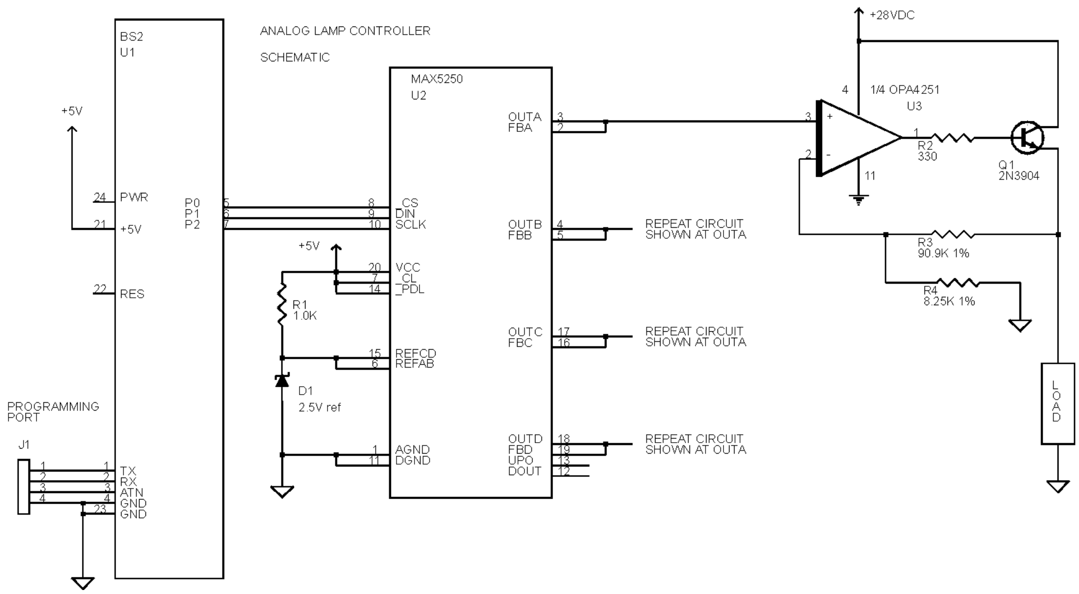
The BS2 can generate analog signals with the PWM command. But this command cannot be executed in conjunction with receiving communication strings (which will occur later in the design cycle). Furthermore, the PWM command cannot generate different analog voltages on four pins at once. So, using the PWM command is out. It looks like we'll have to rely on a stand-alone four-channel DAC.

The MAXIM MAX5250 has plenty of resolution (10 bits), as well as a serial peripheral interface (SPI) which conserves BS2 I/O lines and makes use of the SHIFTOUT command for simplified software control.

The MAX5250 has some internal feedback connections which allow the designer to access some internal gain circuits. For this design, we can eliminate eight resistors by tying the feedback pins directly to their associated analog outputs. See Figure 59.1 for a better understanding of where these pins are located. Also, by using a 2.5V reference diode (such as the National Semiconductor LM4040BIM-2.5), you can squeeze a little more accuracy out of this design. What you're left with is a 10-bit DAC with an analog output ranging from 0-2.5VDC. This translates to 2.5V/1024bits, or 2.44mV/bit. This is definitely less than the specified 0-24VDC, but does act as a building block for this system.

The next step in this design is to get the voltage gain up from a maximum of 2.5VDC coming from the MAX5250 to at least the 24VDC required by this design. Tied into the gain concept is a second consideration for this system: feedback. It is possible that your load might change, or fluctuate, after you set your DAC’s output voltage. This fluctuation can affect the output voltage and cause the overall system to perform poorly. Ideally, you would want to continually adjust your output voltage based on changes in your load. You could tie an analog-to-digital converter (ADC) to your output and read the analog signal that your DAC is creating.

Figure 59.1: The D/A amplifier schematic



But this creates a more expensive and very slow feedback system. This problem can be handled in a more intuitive manner.

An op-amp is an ideal element for increasing, or gaining, the voltage potential for any given system. But many small op-amps, particularly surface-mount products, cannot source large amounts of current while spanning large voltage ranges. To optimize the voltage range and current source capability of this system, we'll use the op-amp and NPN configuration detailed in Figure 59.1. The op-amp will drive the base of the transistor continuously until the voltage on pin 2 is equal to that at pin 3. This performs the feedback function mentioned earlier, and at a much faster rate than a digital solution could provide. Because of this feedback, the voltage at the load (emitter of the transistor) is set by the voltage divider created by R3 and R4. The load voltage will be 12 times the voltage generated by the DAC. The voltage divider for this circuit can allow the output of the transistor to reach 30VDC. For this to occur, the supply would have to be increased, and the op-amp selected must accept this increased voltage level (the Burr Brown OPA4251 is a quad, single-supply, op-amp, which can accept up to 36VDC as an upper voltage rail).

In this circuit, the 2N3904 NPN transistor is used as a pass element to provide current to the load and as an element that drops voltage, thus maintaining its emitter voltage at the level required by the DAC's output. For systems requiring more than the specified 10-20mA of current, different transistors can be selected. Transistor packages that can dissipate more power and are designed for heatsinking (such as the TO-220 or TO-3) can also be used as pass elements and can significantly increase the current source capability. The base resistor may also be adjusted based on the Beta value of any selected transistor, and can be used to limit current in shorted loads.

This system consisting of the DAC, op-amp, and transistor configuration is relatively inexpensive and takes up little room. The step size of voltage control is equal to the DAC's voltage output resolution (2.44mV/step) times that of the ratio of the emitter voltage divided by the voltage at the op-amp feedback signal (this is set by R3 and R4 and equals 12). The end result is that each voltage step requested by the BS2 will equal 2.44mV*12, or 29mV, at the emitter of the transistor. Over a range of 24V, there are about 820 discrete steps. The ratio of R3 and R4 could be further adjusted to force the entire span of DAC steps (1024). You would do this by setting the R3 and R4 voltage divider to 9.6 (instead of 12).

The Software

The software required to interface to the MAX5250 is quite straightforward. The SHIFTOUT command is used to generate a saw-tooth wave on each of the DAC's output pins. The first four calls to the WRITE_TO_DAC subroutine load the new DAC output value into the appropriate buffer registers. Finally, the fifth call to the WRITE_TO_DAC subroutine updates all of the output registers and causes the desired voltage to appear at the DAC output pins. Figure 59.2 shows the software in action.

One bit of information can help to clarify the code. Each DAC in the MAX5250 has a separately addressed buffer register. The voltage value is loaded into this register. In order to access the correct DAC, you must also send an address that is associated with a particular DAC. The exact format that the MAX5250 requires can be located in the data sheet for the part (which can be downloaded from www.maxim-ic.com). In the WRITE_TO_DAC subroutine, the desired voltage is left rotated two bit places and the desired DAC address is added to the result. This process places the data into the format required by the MAX5250.

Figure 59.2: Saw-tooth wave and amplified output

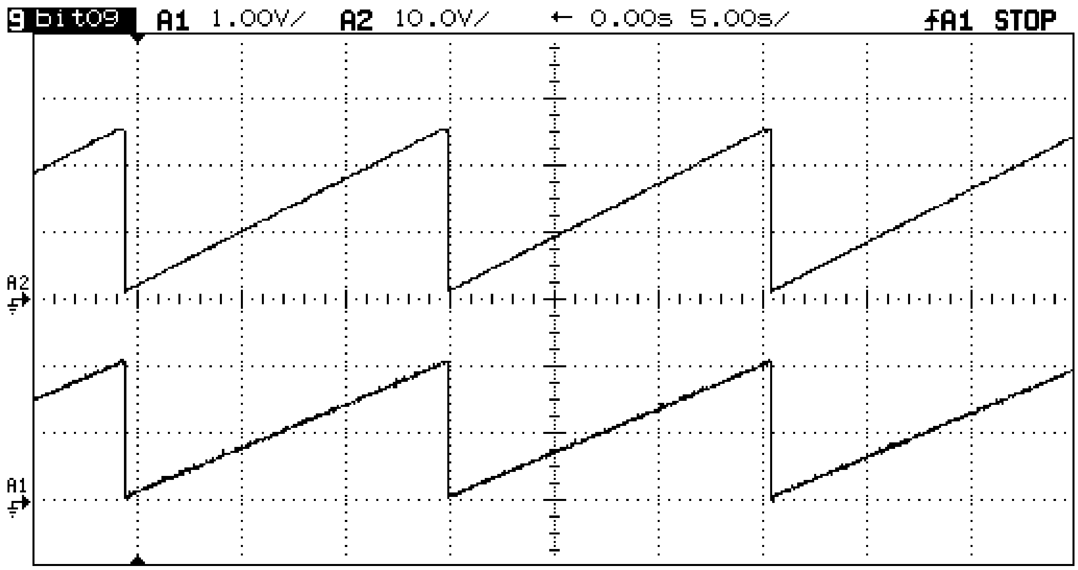


Figure 59.2 displays the output of the DAC on A1, and the amplified output at the emitter of Q1 on A2. Sending 820 consecutive 30mV steps generates the saw-tooth wave. The period of this wave is quite slow, but could be sped up by changing the step size associated with each DAC update. The code in Program Listing 61.1 sends the same value to all four of the DACs in the MAX5250. You could just as easily send four different values.

In Closing

This analog output system is highly accurate and very versatile. Furthermore, it uses very few of a BASIC Stamp's resources. Changing the DAC, op-amp, or transistor can provide a wide variety of cost savings, and/or performance increases for your particular system. This kind of configuration can be used for signal generation for triangle-, square-, or sinewaves. You may even place smoothing filters between the DAC and the op-amp to generate "clean" sinewaves at your output.

Goodbye

This will be my last article for Nuts & Volts, and it has been very enjoyable interacting with all of the Stamp enthusiasts that I have been in contact with. Unfortunately, being part of a growing company requires that I devote my resources to internal projects. It is also a good idea to get some fresh blood into the Stamp Applications cockpit every now and then, so that new innovative ideas can come to light.

In closing, I would like to thank the folks at Nut & Volts, and the readers of this column, for the opportunity to contribute to the Stamp community. I would also like to thank Parallax for their support, and for the excellent products that they provide.

```
'Program Listing 59.1: DAC_AMP.BS2
'DAC_AMP.BS2 writes to the MAX5250 4 channel 10-bit DAC
'generating a saw-tooth wave

       CSpin          CON     0       'DAC chip select
       SDOpin         CON     1       'Data out from BS2
       CLKpin         CON     2       'SPI clock pin

       DACaddr0       CON     $1000   'DAC 1 address
       DACaddr1       CON     $5000   'DAC 2 address
       DACaddr2       CON     $9000   'DAC 3 address
       DACaddr3       CON     $D000   'DAC 4 address
       DACwrite       ON      $4000   'DAC write data to outputsz

       DACreg         VAR     WORD    'DAC voltage value register
       DACloc         VAR     WORD    'Location for voltage value
       SPIreg         VAR     WORD    'Register sent with SHIFTOUT

       DACreg  =      $0000           'Default to 0V out
       HIGH           CSpin           'De-select DAC
       PAUSE          500


START:
       FOR            DACreg = 0 to 820
       DACloc         = DACaddr0      'Load DAC zero
       GOSUB          WRITE_TO_DAC
       DACloc         = DACaddr1      'Load DAC one
       GOSUB          WRITE_TO_DAC
       DACloc         = DACaddr2      'Load DAC two
       GOSUB          WRITE_TO_DAC
       DACloc         = DACaddr3      'Load DAC three
       GOSUB          WRITE_TO_DAC
       DACloc         = DACwrite      'Update DAC outputs
       GOSUB          WRITE_TO_DAC
       NEXT                           'Get next voltage value
       GOTO           START

WRITE_TO_DAC:
       SPIreg         = DACreg<<2     'Shift voltage value by two bits
       SPIreg         = SPIreg+DACloc  'Add address to voltage value
       LOW            CSpin            'Select DAC
       SHIFTOUT       SDOpin,CLKpin,msbfirst,[SPIreg\16]
       HIGH           CSpin
       RETURN
END:
```