D/A Converters

ECE 376 Embedded Systems

Jake Glower - Lecture #17

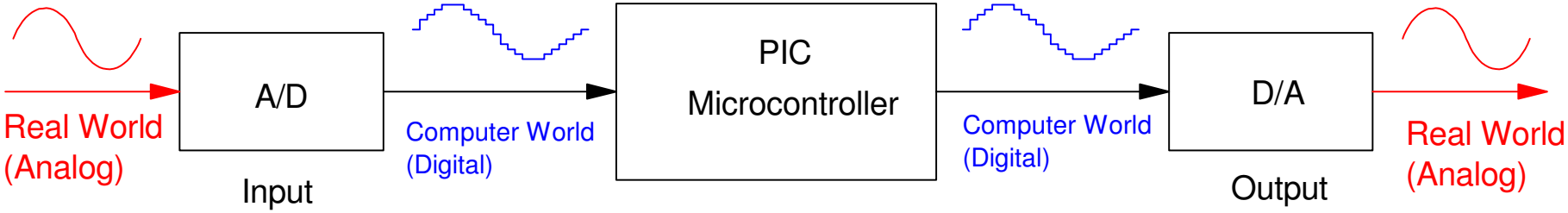
Please visit Bison Academy for corresponding
lecture notes, homework sets, and solutions

D/A Converters

A/D: Allows you to input analog signals to a PIC

D/A: Allows you to outout analog signals from a PIC

- Output a sine wave to drive a speaker (rather than a square wave)
- Output other periodic waveforms, creating an electonic cello, violin, tuba, etc
- Control the speed of a DC motor
- Control the brightness of an LED, and
- Control the speed of an AC motor to name a few.



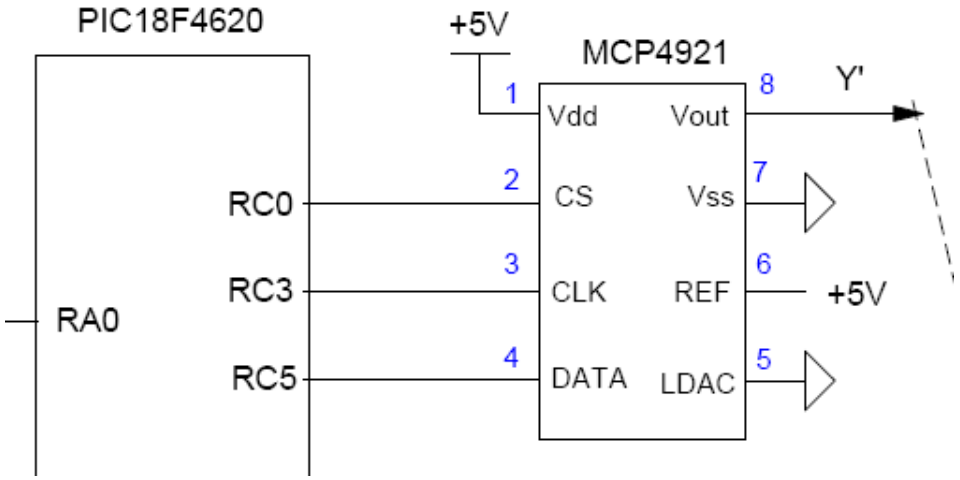
D/A with a PIC Microcontroller

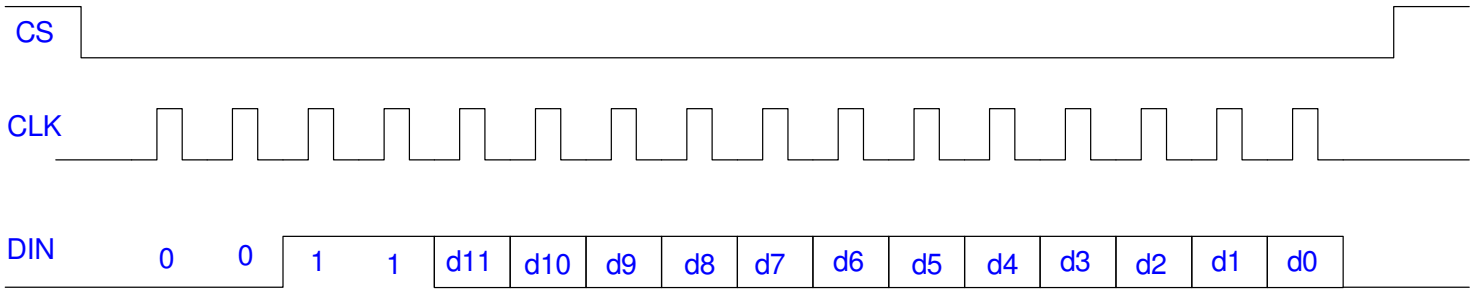
MCP4921: External D/A chip

- PIC doesn't come with a D/A

Procdure: Send 16-bits of data

- Pull CS low (start of message)
- Send 0011
  - This sets the output to 0% to 100% of Ref.
- Send 12 bits of data, MSB first
  - 0 (0x000) outputs 0V
  - 4095 (0xFFF) outputs 5V
- Pull CS low (end of message).





Software: D/A Driver

```
void D2A(unsigned int X)
{
   unsigned char i;
   TRISC0 = 0;
   TRISC3 = 0;
   TRISC5 = 0;

// Add 0011 to the first four bits to set up the D/A
   X = X & 0x0FFF;
   X = X + 0x3000;
   RC0 = 1;
   RC3 = 1;

// CS goes low to select the D/A chip
   RC0 = 0;

// Send out 16 bits of data
   for (i=0; i<16; i++) {
      if (X & 0x8000) RC5 = 1; else RC5 = 0;
      RC3 = 0;
      X = X << 1;
      RC3 = 1;
      }

// CS goes high to terminate the communicaitons
   RC0 = 1;
}
```

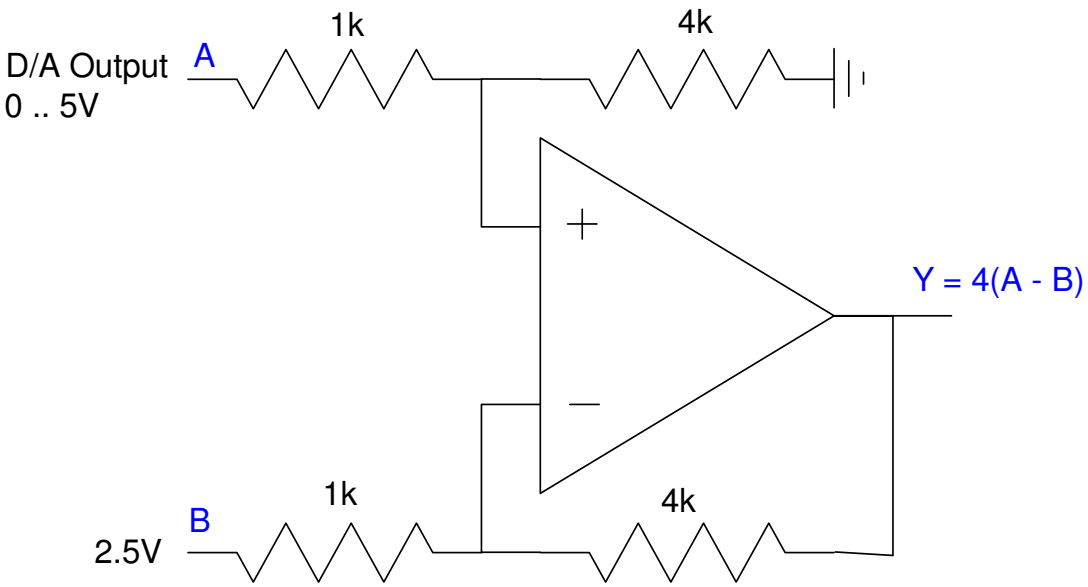
Electronic Function Generator

Binary Output: Square wave

- Creates a harsh sound

Analog Output: Square / Sawtooth / Triangle / Sine wave

Step #1: Convert 0..5V to -10V .. +10V



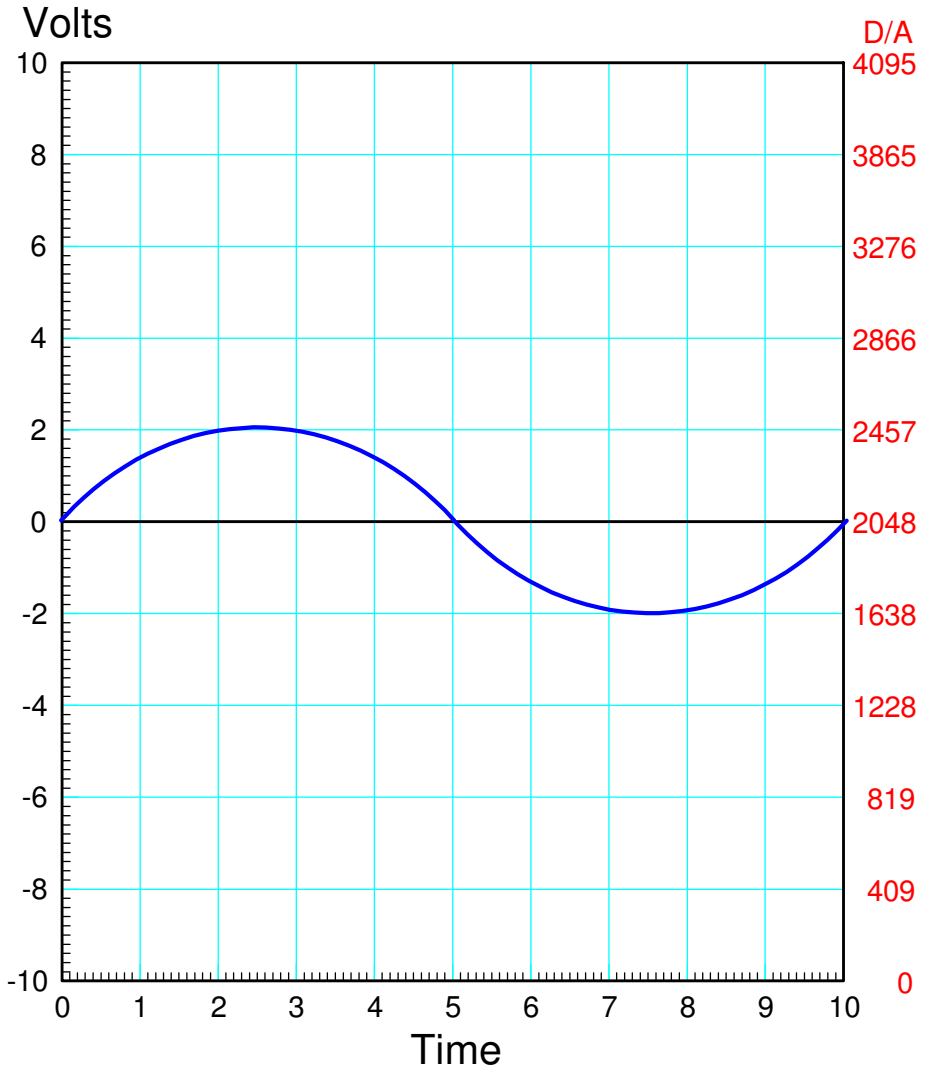
Step #2: Create waveforms

- Assume the output goes -2V to +2V

The output voltage is related to the number you send to the D/A as

$$Volts = \left(\frac{D/A}{4095}\right) \cdot 20V - 10V$$

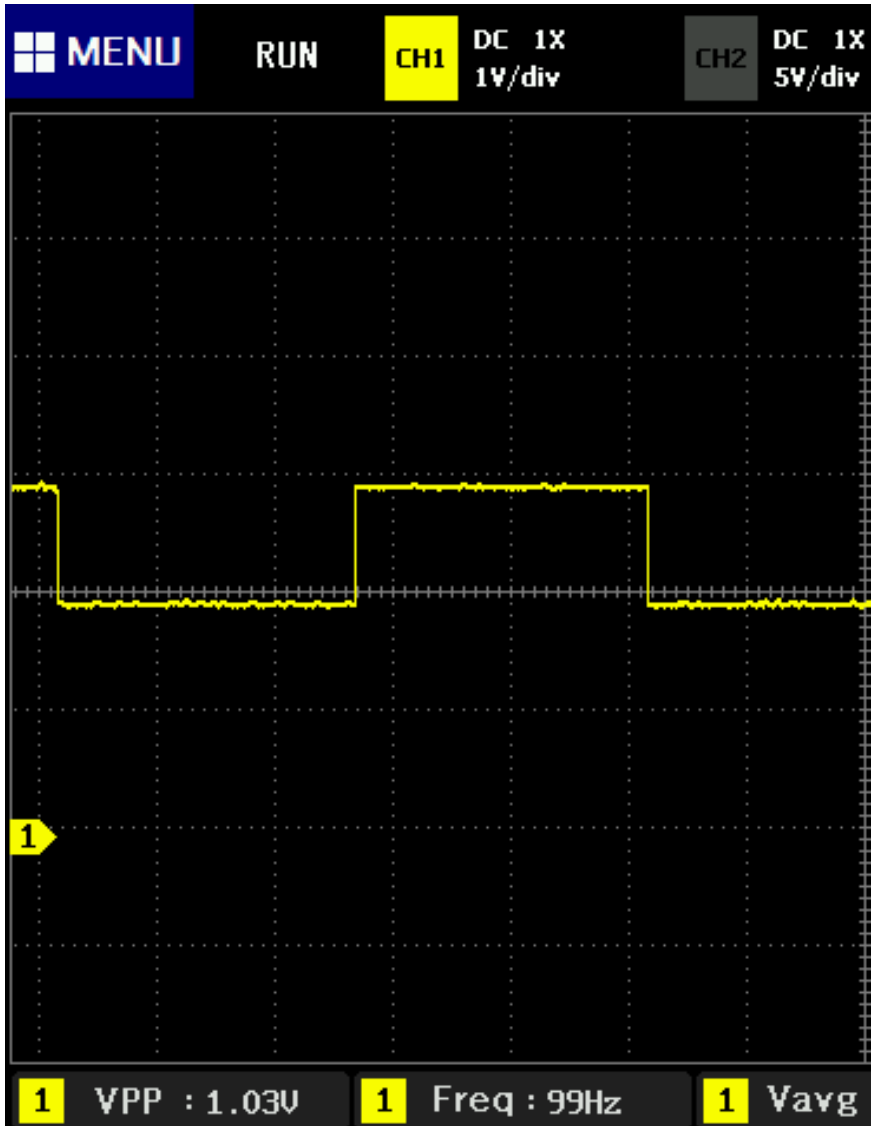
- -2V = 1638
- +2V = 2457



Square Wave

- 100 Hz
- Counting to 620 = 1ms

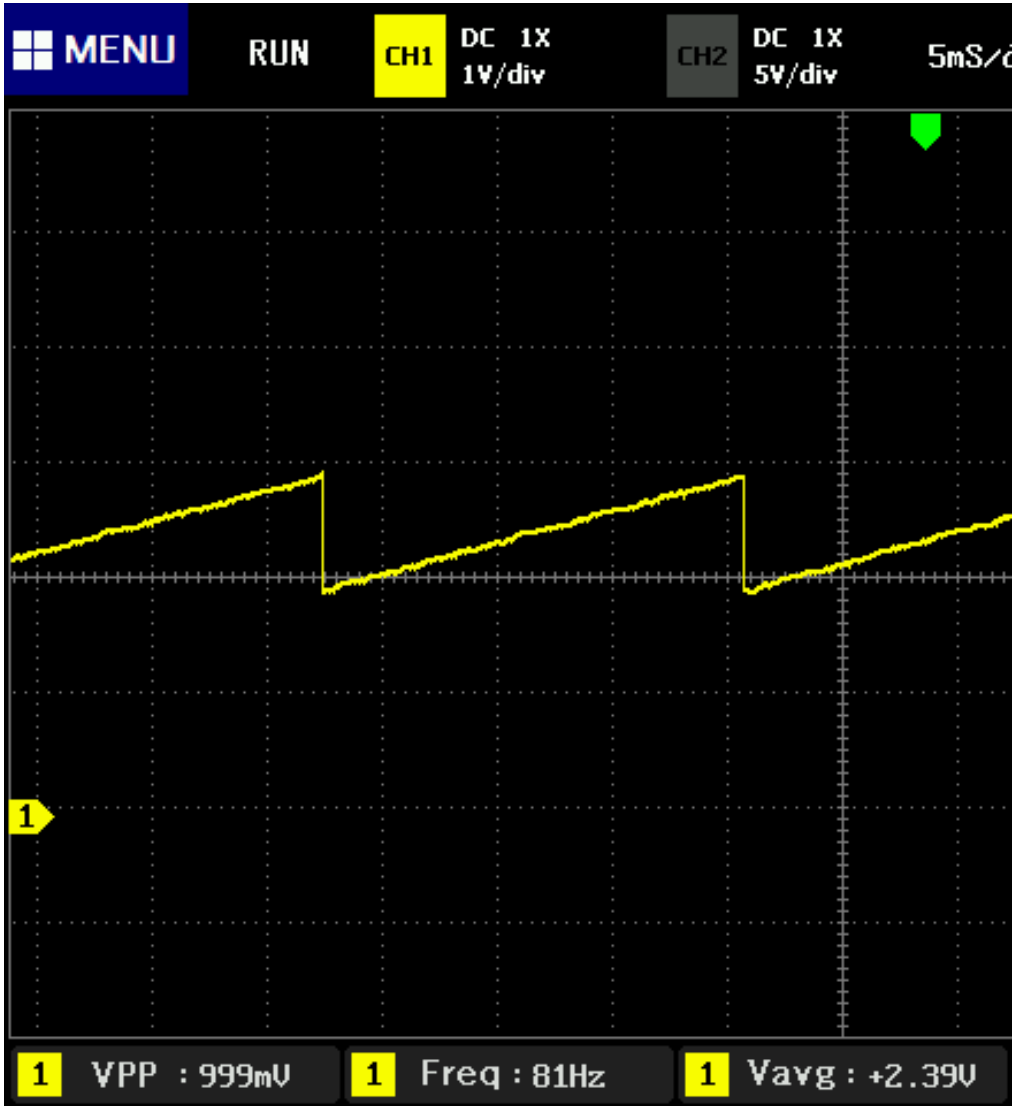
```
while(1) {
   D2A(1638);
   for(i=1; i<3100; i++);
   D2A(2457);
   for(i=0; i<3100; i++);
   }
```



Sawtooth Wave

- 81 Hz

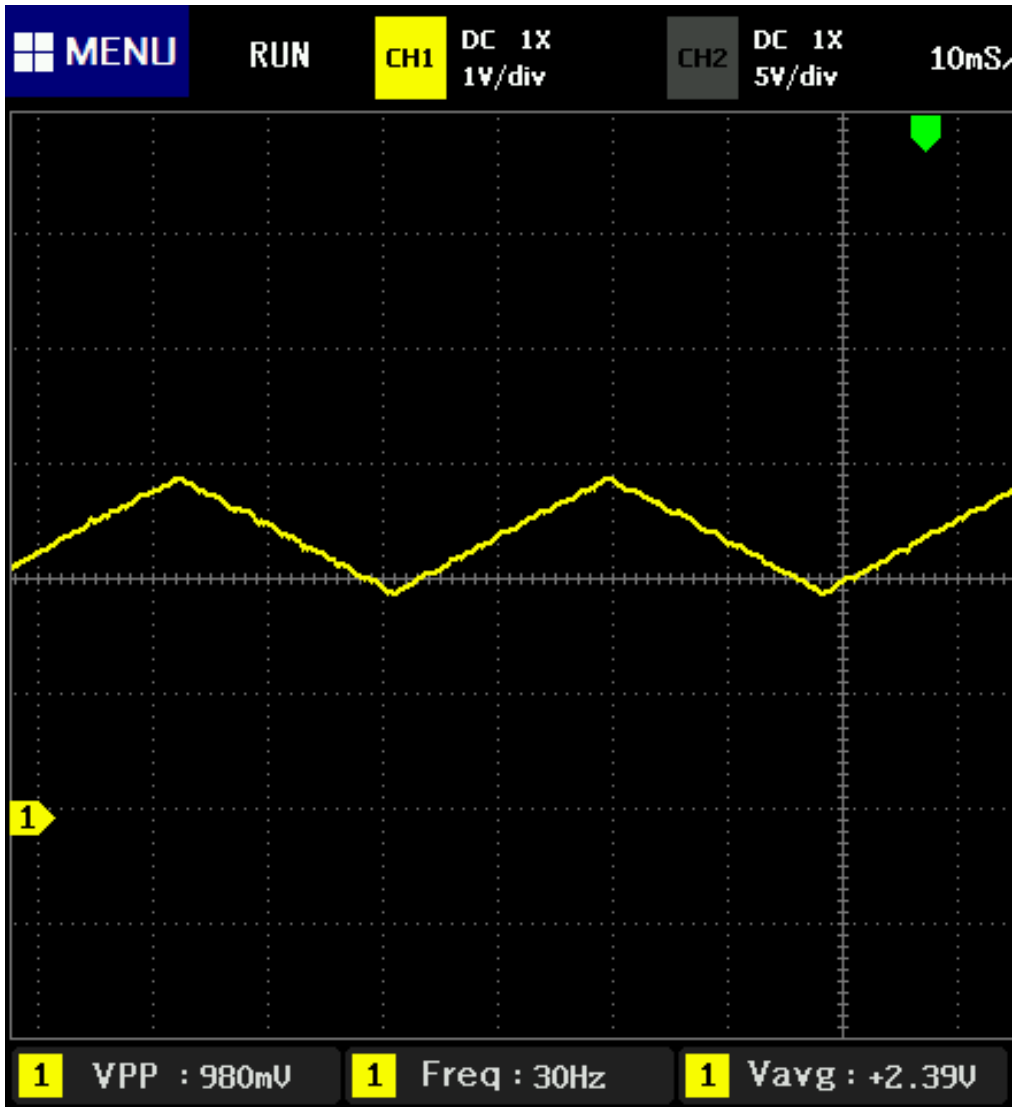
```
x = 1638;
while(1) {
   D2A(x);
   x = x + 10;
   if(x > 2457) x = 1638;
   for(i=0; i<100; i++);
   }
```



Triangle Wave

- 30 Hz

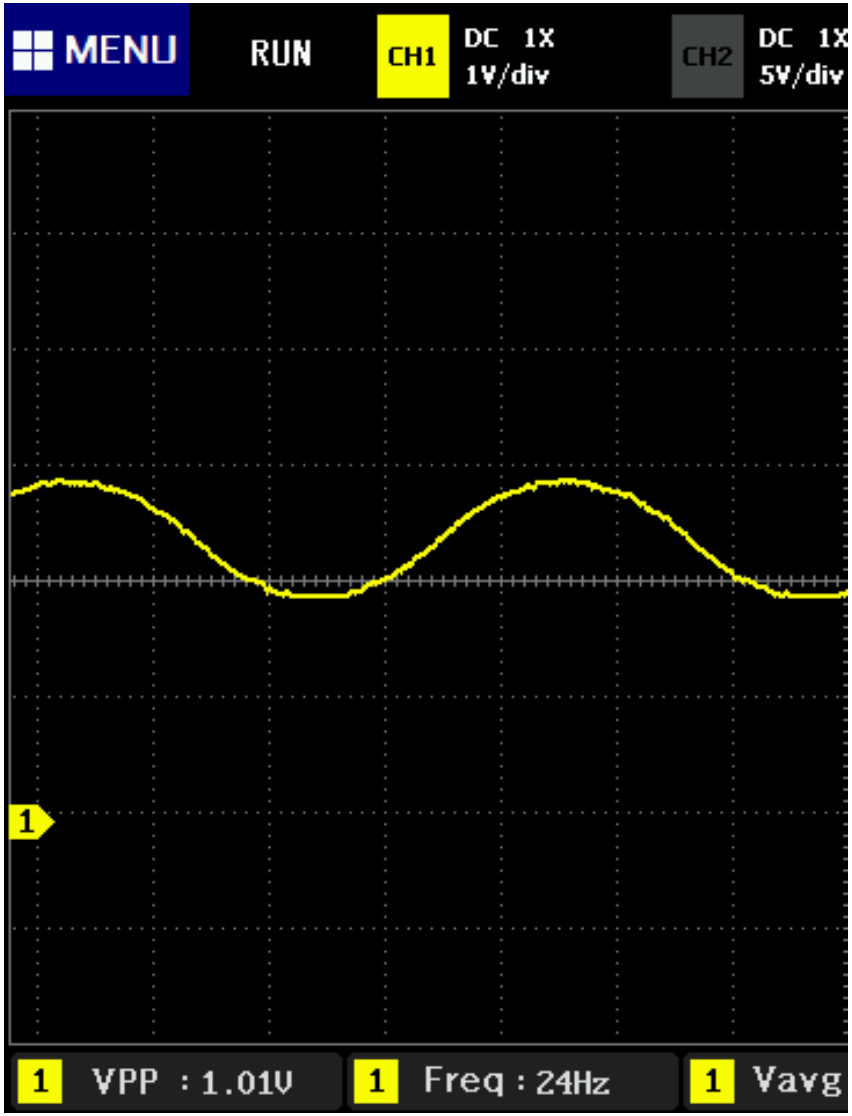
```
x = 1638;
dx = +10;
while(1) {
   D2A(x);
   x = x + dx;
   if(x > 2457) dx = -10;
   if(x < 1638) dx = +10;
   for(i=0; i<100; i++);
   }
```



Parabolic Sine Wave

- of Mutant Ninja Warrier fame
- Season #1, episode #2

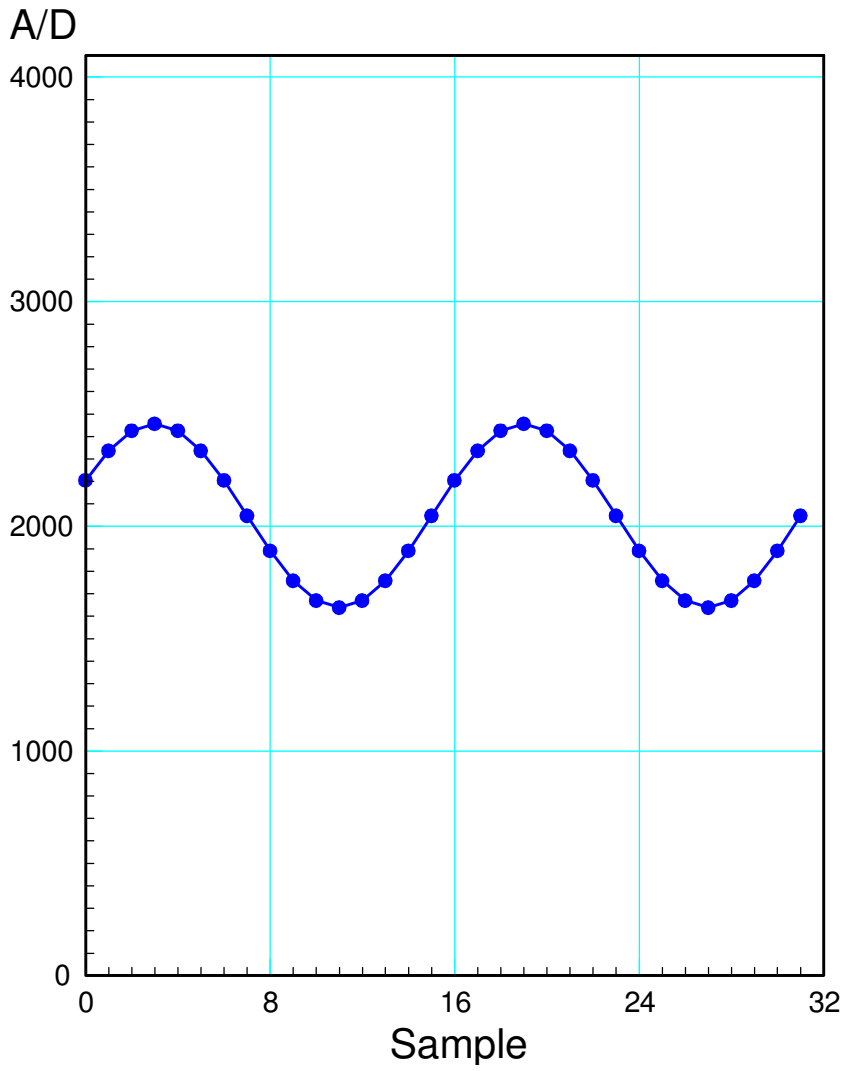
```
//Parabolic Sine Wave
   x = 0;
   dx = 0;
   while(1) {
      x = (x + 1) % 64;
      if(x == 0) dx = dx ^ 1;
      if(dx) y = 2048 + (2 * x * (64-x))/5;
      else y = 2048 - (2 * x * (64-x))/5;
      D2A(y);
      for(i=0; i<100; i++);
      }
```



Sine Wave:: Look-Up Table

- Generate from matlab

```
t = [1:16]' / 16 * 2 * pi;
y = sin(t) * 409 + 2047;
y = round(y);
     2204.
     2336.
     2425.
     2456.
     2425.
     2336.
     2204.
     2047.
     1890.
     1758.
     1669.
     1638.
     1669.
     1758.
     1890.
```

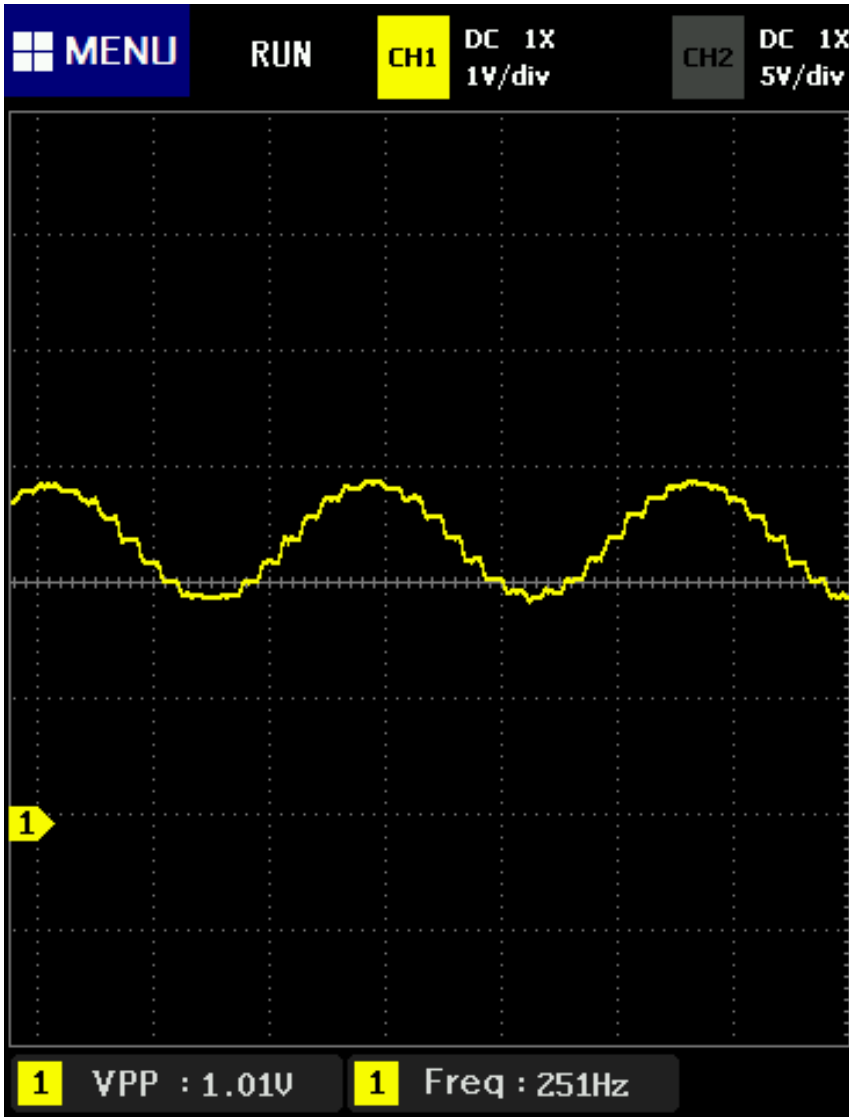


To play a 220Hz sine wave,

One cycle = 1/220 second (4.54ms)

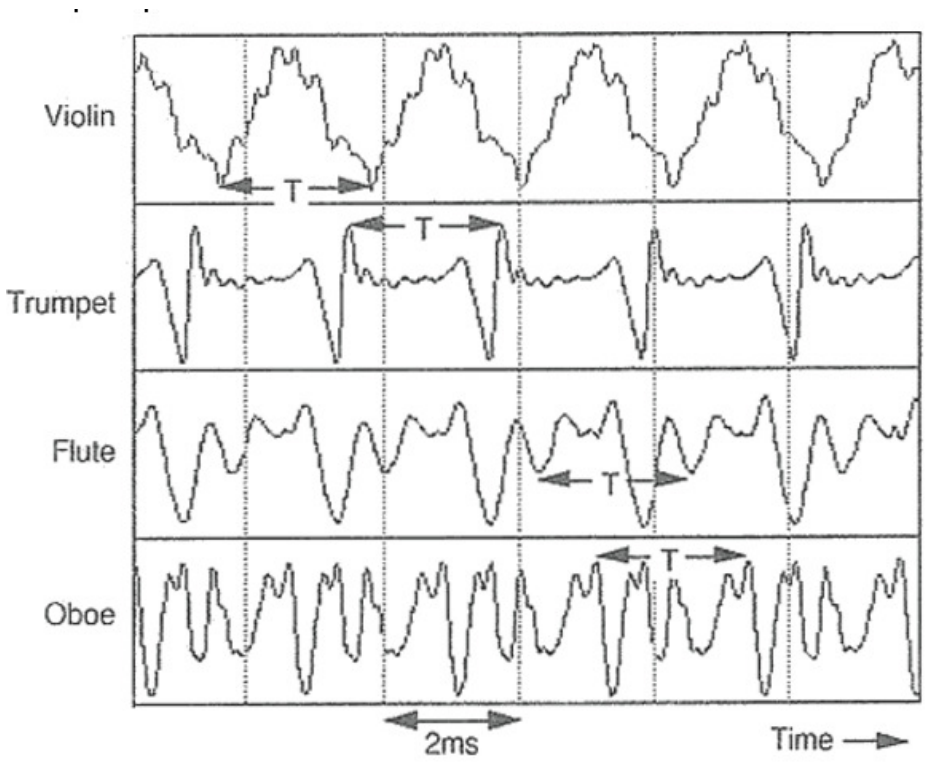
Each D/A call = 4.54ms / 16 = 286.1us

```
const unsigned int TABLE[16] = {2204, 2336,
2425, 2456, 2425, 2336, 2204, 2047, 1890, 1758,
1669, 1638, 1669, 1758, 1890, 2047};

// in the main routine
   unsigned int i, j;

   while(1) {
      i = (i + 1) % 16;
      D2A(TABLE[i]);
      for(j=0; j<177; j++);
      }
```



Electronic Tuba

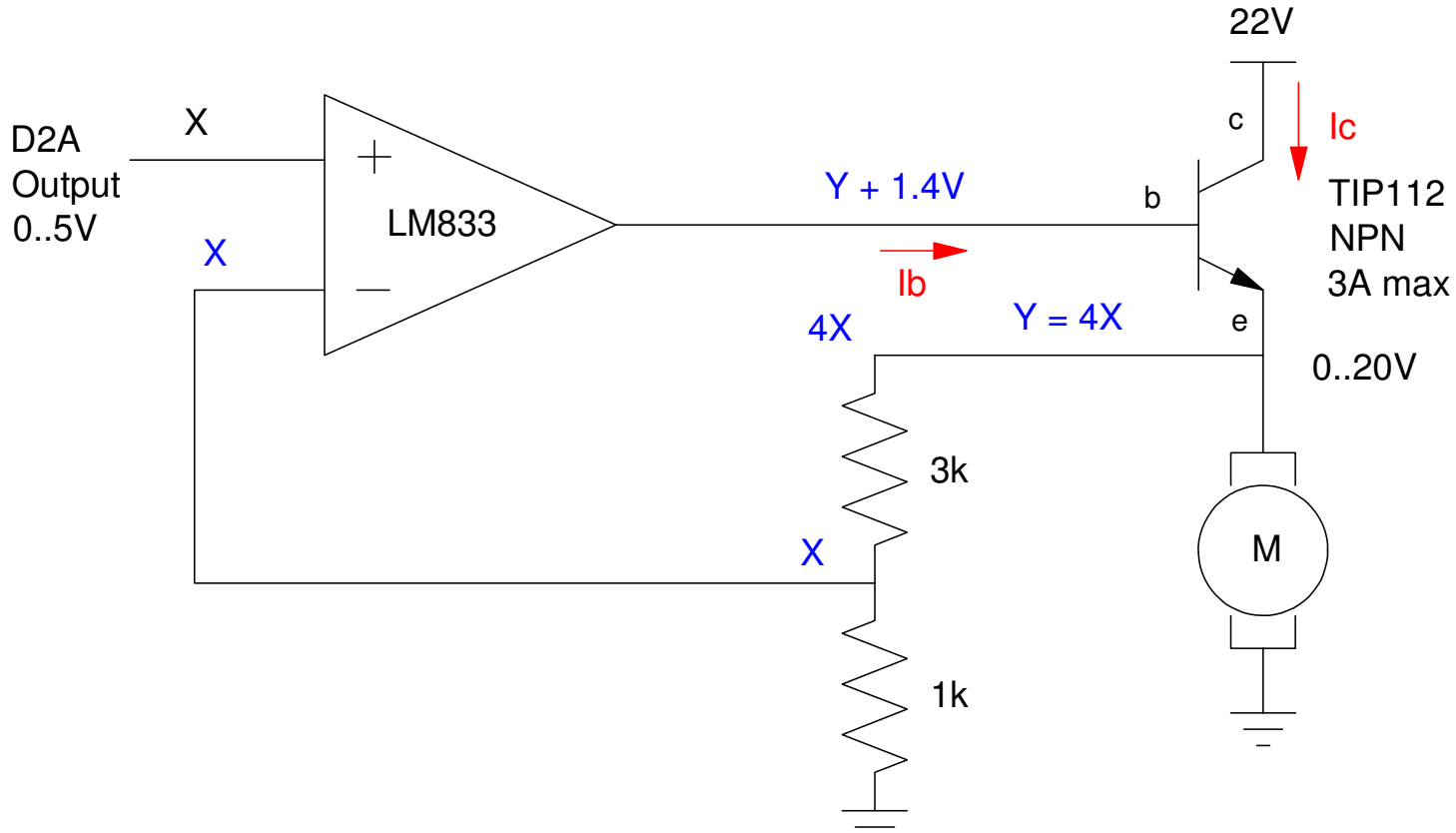
Note that if you change the look-up table, you change the signal sent to the speaker. The signal determines what type of instrument you're mimicking:



http://www.feilding.net/sfuad/musi3012-01/html/lectures/009_hearing_IV.htm

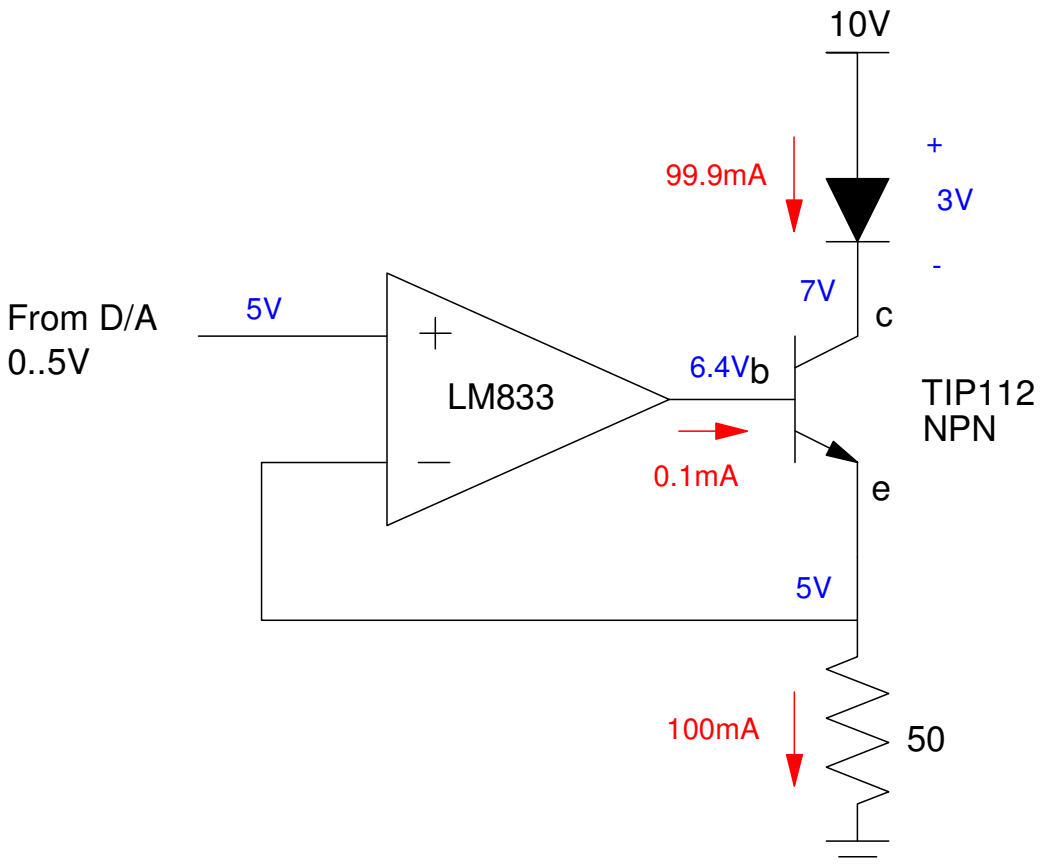
DC Motor: Speed Control

- Voltage = Speed (approx)
- 0V .. 20V with 4095 steps



Current Source:

- Control the brightness of an LED (current = lumens)
- Control the torque of a DC servo motor (current = torque)



DC Servo Motor: Speed Control

AMD 30A8V

- Voltage range: +20V to +80V
- Current Limit: Up to 20A
- Higher efficiency (data sheets say 90%)
- Protection circuitry (doesn't die as easily)

Advanced Motion Controls 30A8V Brush Type PWM Servo Amplifier

Be the first to write a review.

Condition: Used

Quantity: 1 More than 10 available

79 sold / See feedback

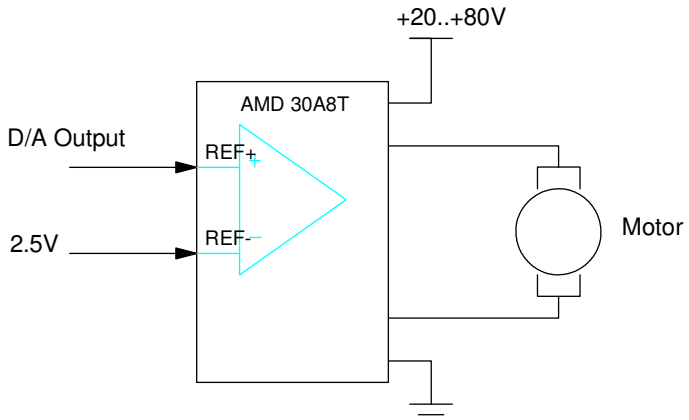
Price: US $39.99

Buy It Now

Add to cart

Just connect

- REF+ D/A output (0..5V)
- REF- 2.5V
- Adjust the gain
  - +30V at+5V,
  - -30V at -5V



BLDC Motor: Speed Control

- 3-Phase AC Synchronous Motor
- Use a motor driver ($14 on ebay for 380W)

380W 3 Phases Brushless Motor Controller Board(No/Without Hall Sensor)BLDC D0Q6

Be the first to write a review.

Condition: New

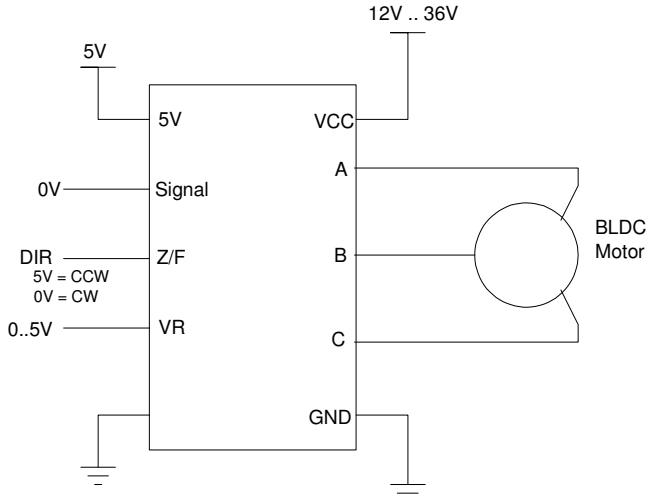
Bulk savings: Buy 1 $13.98/ea | Buy 2 $13.28/ea | Buy 3 $13.14/ea

These boards

- Take an analog input (0..5V), and
- Generate 3-phase AC waveforms
- With a frequency (i.e. speed) proportional to the input voltage

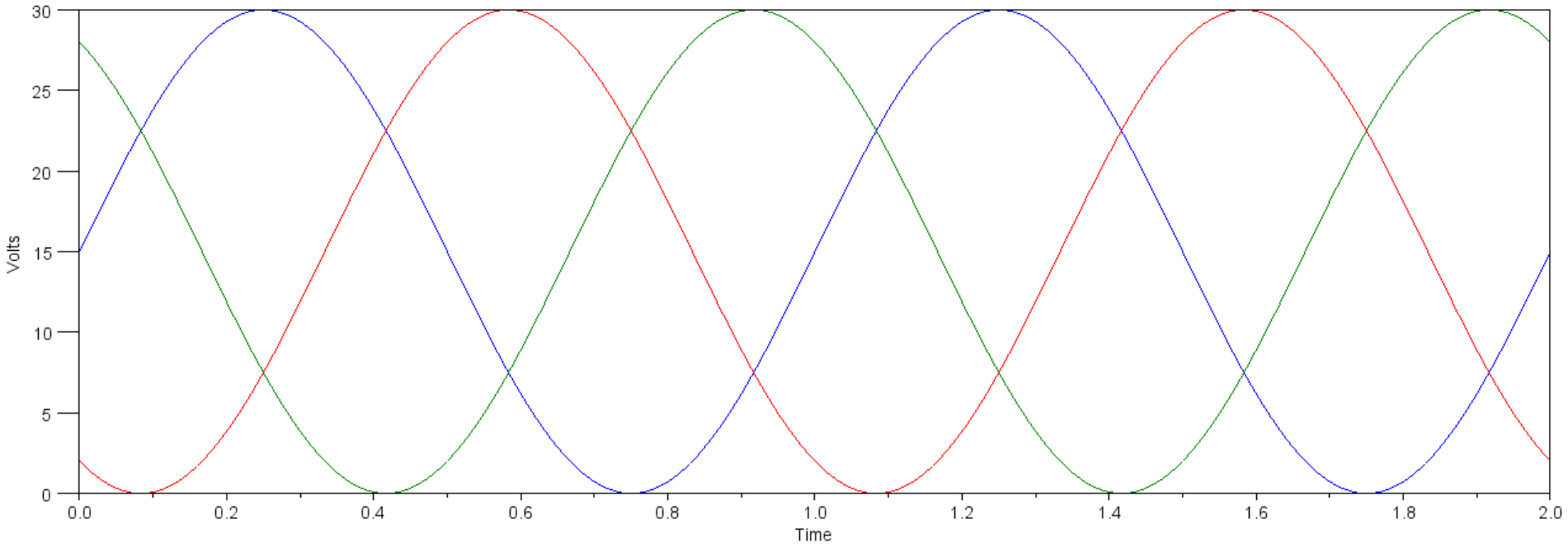
D/A outputs 0..5V

- Controls the speed
- Also sets the direction (another binary input)



Output Waveform

- Frequency = Speed



Va, Vb, Vc

Vab, Vbc, Vca remove the DC offset