MSE491 – D200 2024

Advanced Analog Electronics

Prof. Patrick Palmer

Experiment 1

The R-2R Ladder with an Op-Amp Buffer/Amplifier and Low Pass Filter

The Digital to Analog converter is an important feature of many systems. The Raspberry Pi PICO does not posses a peripheral for this function, as it has a very fast clock and it is supposed that the analog output may be obtained using principles of PWM. However, in this experiment you will lean about the R-2R ladder and begin a hands-on journey into precision electronics.

Learning Outcomes

1) Use of Thevenin and superposition to analyse a complicated circuit.
2) Consideration of impedances looking in and looking out
3) An Op-Amp Buffer/Amplifier
4) Single Rail Op-Amp circuits with input and output operating down to ground
5) Op-Amp Datasheets and input considerations
6) Waveform synthesis
7) Basic Op-Amp Filter design
8) Use of an oscilloscope to view a repetitive hard to trigger waveform
9) Simulation of circuits (LTspice)

Section A: Raspberry Pi Pico

The Raspberry Pi Pico W is shown in Figure 1.

Figure 1 The Raspberry Pi PICO W

‘The Pico W also uses an onboard buck-boost SMPS to generate the 3.3 V required for the RP2040 and external circuitry. This SMPS allows a wide range of input voltages to be used, 1.8 V to 5.5 V, which can come from a variety of sources, single lithium-ion cell, USB, or even three AA batteries in series.’

The Raspberry Pi Pico can be conveniently programmed using the Desktop application Thonny and the interpreted language MicroPython, loaded onto the Pico.

If you are starting with a new Pico board, you will need to solder pins to the board and load Micropython. Instructions for doing so can be found here:

https://microcontrollerslab.com/getting-started-raspberry-pi-pico-thonny-ide/

Make sure your RPI Pico W is connected to your computer using a proper USB data cable (e.g. branded, fat, but not skinny). Start Thonny (if it is not running) and check the bottom right corner and see if Micropython is listed. If so, select it as your ‘target’. Then try the following program to perform the ‘Hello World’ function.

Code:

```
'''Blink the on board LED
Pico W pinouts only
Patrick Palmer Jan 2024
'''

from machine import Pin, Timer
led = machine.Pin("LED", machine.Pin.OUT)
timer = machine.Timer()

led.on()
def blink(timer):
   led.toggle()

timer.init(freq=10, mode=Timer.PERIODIC, callback=blink)
```

You can run the program from the Thonny interface and you should see the activities in the bottom section of the thonny interface.

Save the program, if it is working, on your computer, as you will need it later.

Note: you can also save the program on the RPI Pico W. If you give it the name main.py, it will run after a reboot.

Section B: R-2R Ladder

On a dual Breadboard, build the R-2R ladder circuit shown in Fig. 2. Leave some room at the end of the breadboard for the LM358 Op-Amp Amplifier/Buffer circuit. Using 1k resistors, the R-2R arrangement is easy to develop correctly.

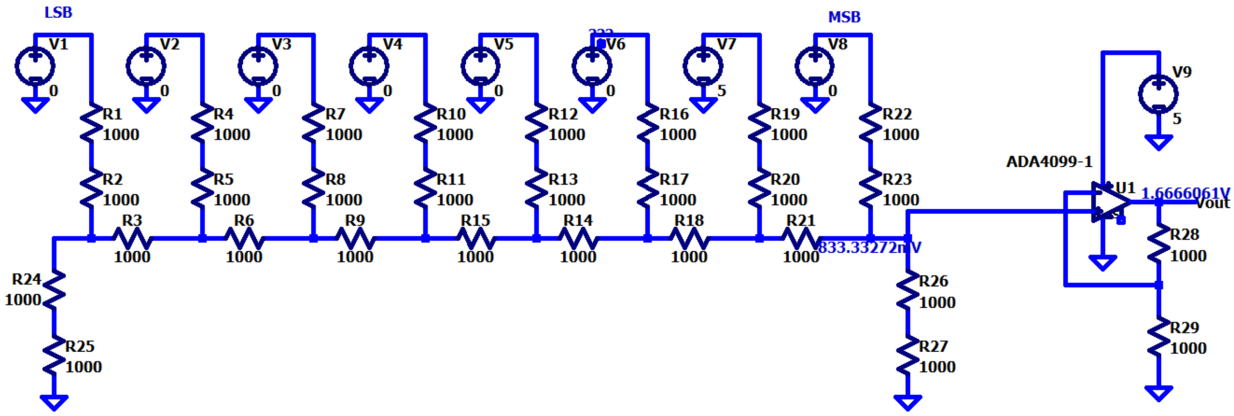

Figure 2 R-2R circuit with a Buffer/Amplifier

Remember that the Breadboard has strips of holes sharing the same 'node', in two banks, either side of the gap. So think carefully about one 'cell' before starting to build your circuit.
It lends itself to two halves horizontally, with the Pico on a separate Breadboard (two are provided, side by side).

Connect 8 General Purpose (GP) lines from the Pico to the R-2R ladder as inputs. Take care to use the Pico W pinout guide on Canvas. Make a note of the pin numbers and which end is LSB and which is MSB. Connect your R-2R ladder ground to the Pico W ground.

- ⇒ Copy the Hello World code and adapt it to produce a slowly toggling output on the MSB GP pin. Measure the output voltage appearing at the output of the R-2R Ladder using the oscilloscope. Make a note of the high and low voltage values. Repeat this for the next bit, and so on until the you have measured all 8 bit voltage values as they appear at the output of the R-2R ladder.
- ⇒ Calculate or Simulate for the voltages at the output of the R-2R Ladder for each bit and compare your measured values to the calculated values. Do you notice any trend.
- ⇒ At the output of the R-2R Ladder, add a load of 1k. Try a few 'bits' and note down the values of the output voltage measured. What do you notice about these values?

You should also run the simulation provided, using pulse voltages in place of the Data pins.

Section C: Op-Amp Buffer circuit

To drive a load in a well characterized manner, it is typical to add aa 'Buffer' op amp. The LM358 is an inexpensive but very capable general purpose Op-Amp, which may be driven to zero at the inputs, and gives near zero at the output even when used in 'single ended' supplies. Since the Pico W is powered by the 5V USB, you may power your LM358 using the 5V out pin on the Pico W (see pinout guide). Using 1k resistors set up the LM358 as a Buffer with a gain of 2, as shown in Fig. 3, connecting it's free input to the output of your R-2R ladder.

- ⇒ As above, for a few bits measure the output voltage of the Buffered R-2R ladder. Are these values what you would expect?
- ⇒ For a zero digital output, what is the output of the Buffered R-2R ladder?
- ⇒ Consult the LM358 Datasheet and consider it's capabilities. Can the output accuracy be improved, noting that the inputs may draw up to 500nA? If you already taken into account this factor then remove your remediation and measure the output voltage. If you have not taken into account this factor, consider how you might do so. One solution is to change the values of the Feedback components.
- ⇒ Perform some calculations to determine what you might do (or to explain what you did). Consult with a demonstrator if needed. Try your remedy and note down the new value of output voltage.
- ⇒ Perform a transient simulation using ***LTspice***, and comment on its accuracy.
- ⇒ Set a digital output voltage somewhere around the middle of the range and measure the Buffered output voltage for a varyinge load resistor R_L including steps of 50W, 100Ω, 220, 470 and 1k. What do you notice about the output voltage?

Section D: Arbitrary Waveform Generator

Now that your circuit is performing reliably and accurately, see if you can set it up using micropython to generate a sinewave digitally. Note that Micropython is not a fast language, so your code should try and use simple formulations to develop a sinewave output.

Options which I have found

1. A sine lookup table LUT (preferred to real time calculations).
2. Using it as 2 X 4bit DAC to synthesise the waveform (Truncated Taylor Series)
3. Harmonic elimination – Get rid of the 3^{rd} Harmonic by Quasi-Square wave, using timings
4. PWM of the MSB (Naturally Sampled or elimination of most of 3^{rd} and 5^{th}, by adding pulses)
5. Fourier approach, using multiple harmonic frequency squarewaves as pseudo sinewaves and synthesis of the sine from the fundamental squarewave using Fourier coefficients
6. Flat top and bottom with a uniform ramp in between (mayby 15 levels, so 4 bits)

⇒ Implement your chosen method on the Raspberry Pi PICO. To help trigger the oscilloscope, use a separate I/O pin to produce a square wave (Toggle) or pulse synchonised to your sinewave. You may also add a small capacitor at the output of the R-2R Ladder.

⇒ Filter the output using the low pass filter circuit for the Op-Amp.

⇒ Capture your waveforms before and after filtering! (phone camera or snip tool)

References:

Raspberry Pi Pico W Pinout:

https://www.raspberrypi.com/documentation/microcontrollers/images/picow-pinout.svg

LM358 Datasheet:

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/697/LM158,258,358,2904.pdf

Use of Two DACs

https://www.analog.com/media/en/technical-documentation/application-notes/463837963an322.pdf

Multilevel Inverter

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9001064