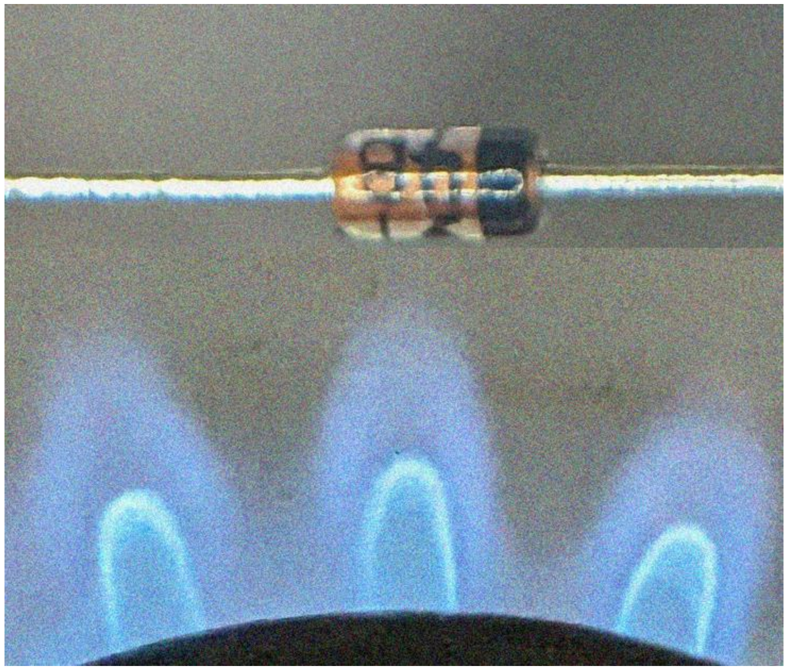
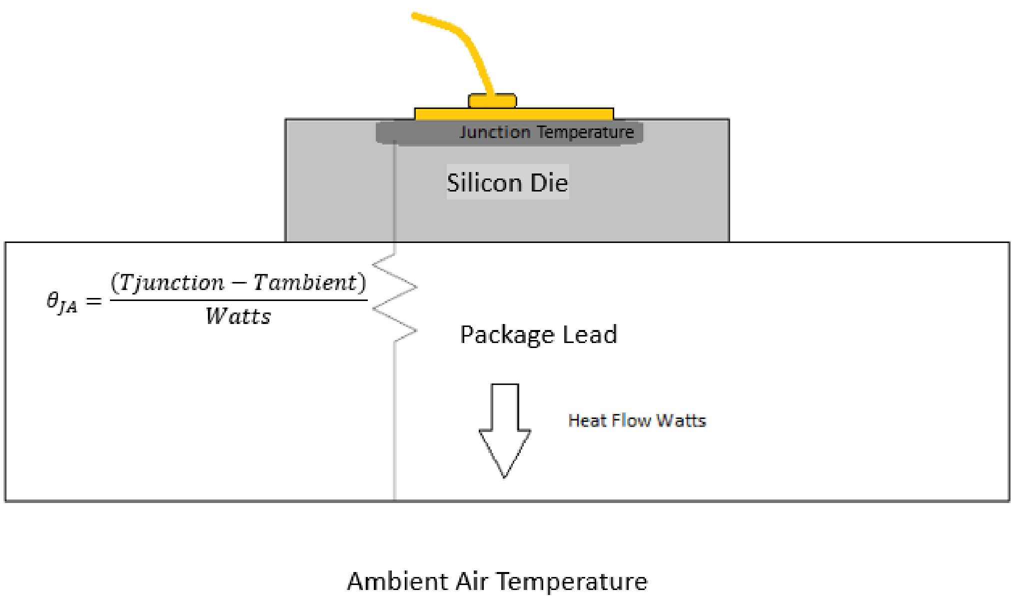
Measure the junction to ambient thermal resistance of a 1N914 diode using your Pico processor.





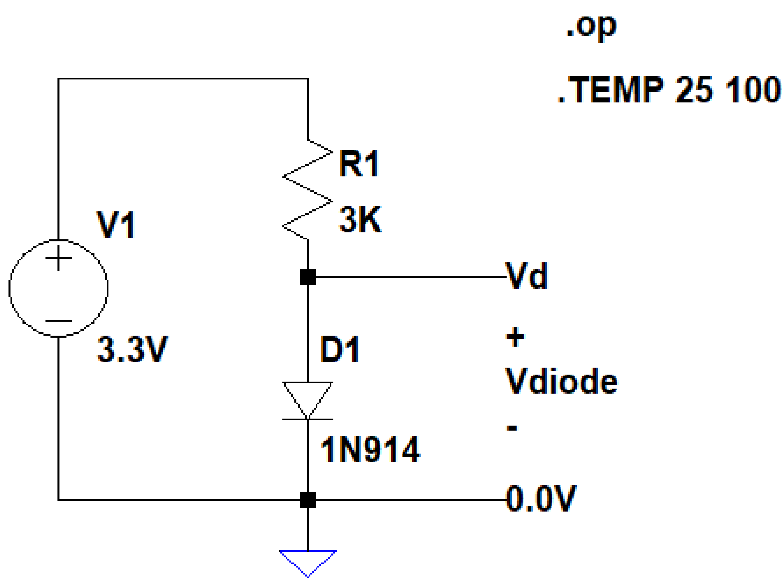
Thermal resistance is measured between two points with heat flow. Heat flow causes the source point to increase in temperature with respect to a sink such as room air. In thermal systems power is analogous to current and temperature is analogous to voltage. Thermal resistance is referred to as theta with the hot and cold reference points. The thermal resistance from the die junction to the ambient air temperature is theta JA. The units are degrees C per Watt.

Y

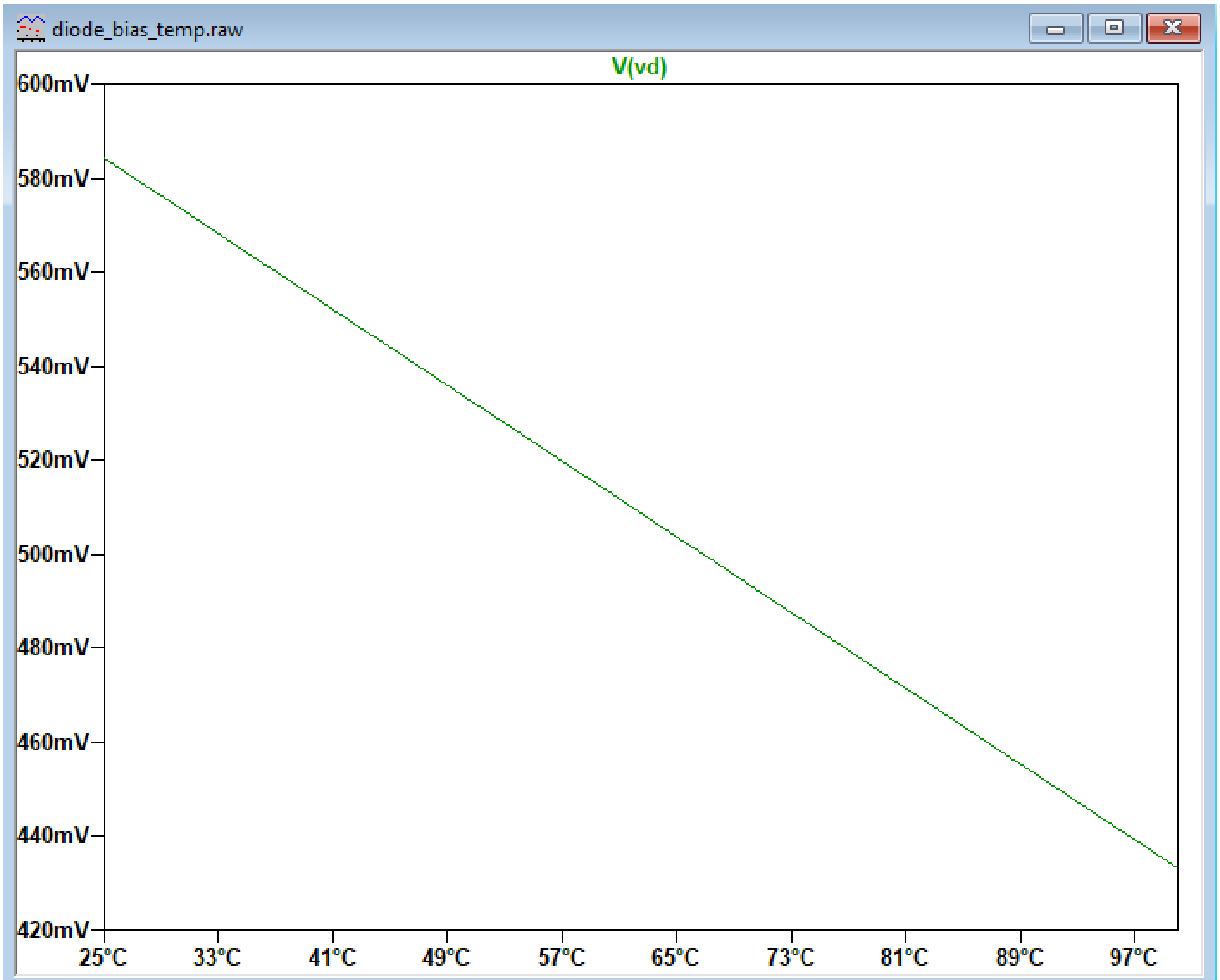
- Knowing the thermal resistance value helps to calculate the die temperature at a given power dissipation.
- The manufacturer specifies the thermal resistance of their products.
    - Measuring thermal resistance teaches you about the technique.
- Silicon die can function at very hot temperatures.
    - Plastic encapsulation, solders and conductive adhesives usually limit the maximum die temperature.
    - Maximum die operation temperature is typically 125 °C although some are much higher.

To find Θ_{JA} (theta Junction to Ambient)

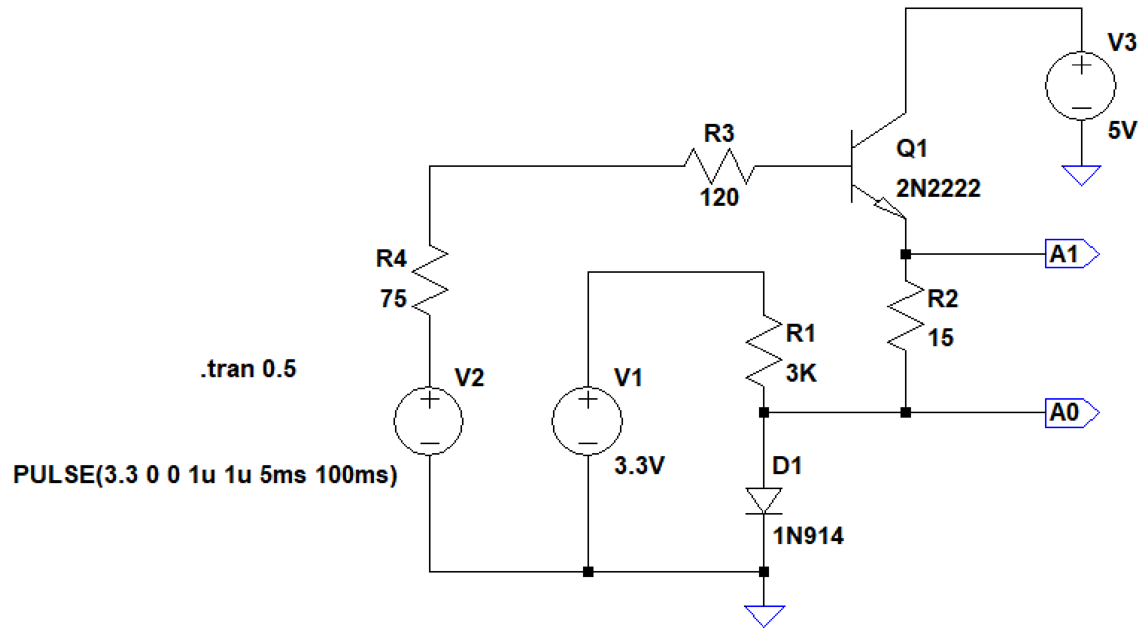
- Measure the change in die temperature with respect to the ambient air temperature.
- Measure the power the die is dissipating.



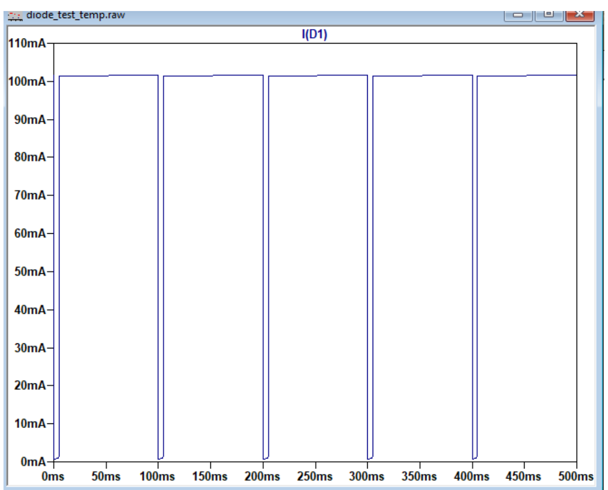
Silicon junction diodes have a well know temperature coefficient of about -2mV per degrees C. Let's test a 1N914 diode model using LTSpice. The diode is biased at about 0.9mA to keep self-heating low.

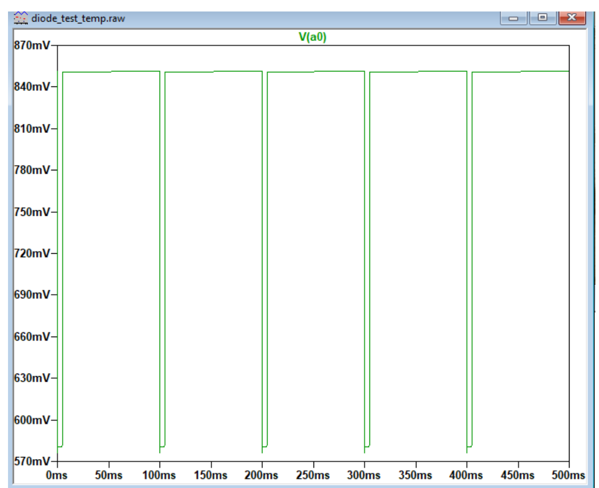


The diode voltage is 584mV at 25°C and 433 mv at 100°C. (584-433)mV/(25-100)°C = -2.01mV/°C



To measure thermal resistance the diode must heat up. Current is increased in pulses using an NPN emitter follower. The current increases to about 100mA, heating the diode. It is turned off for a short period and the diode voltage can be measured to obtain the temperature of the diode die.

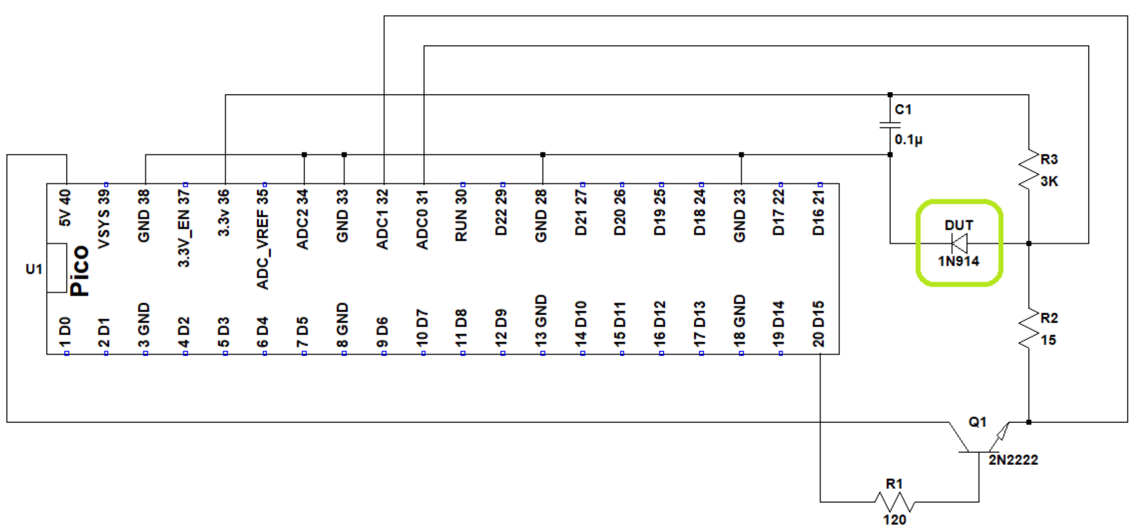




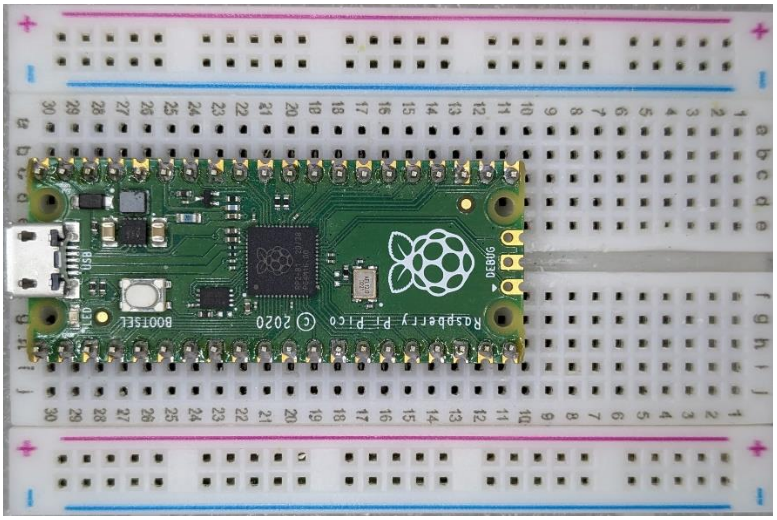
Simulation shows the current pulsing from 102mA down to 0.9mA. The diode voltage is 851mV at high current where the diode heating is at 87mW. When the current drops to 0.9mA, the temperature is measured. This voltage is 580mV. It is the reference cold voltage before the die heats up.

Parts list:

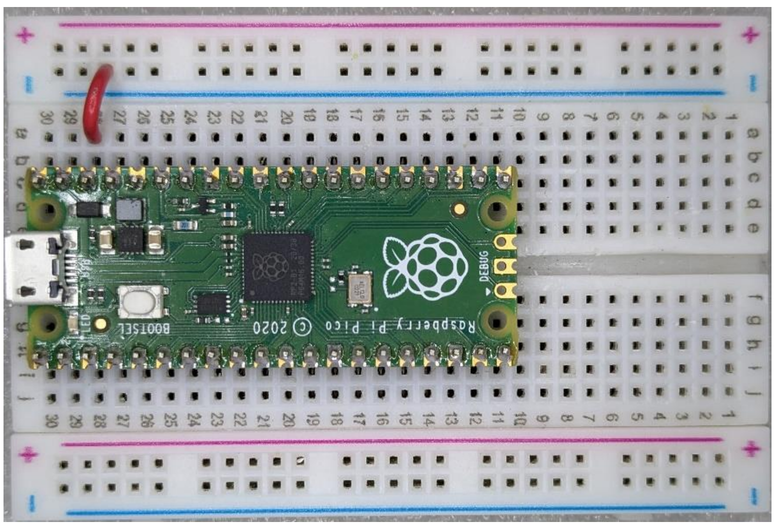
- Raspberry Pi Pico with pins
- 2700 to 3300 ohm ¼W resistor (R3)
- 15 ohm ¼W resistor
- 120 ohm 1/4 W resistor
- 2N2222 or 2N2222A NPN transistor
- 1N914 leaded diode
- 0.1uF ceramic capacitor
- Solderless Breadboard
- 8 – short jumper wires
- 4- long jumper wires
- USB cable
- Computer
- Software
    - The Arduino IDE
    - Earle Philhower's arduino-pico must be installed.
        - Go to the Github web page and follow the instructions.
        - https://github.com/earlephilhower/arduino-pico



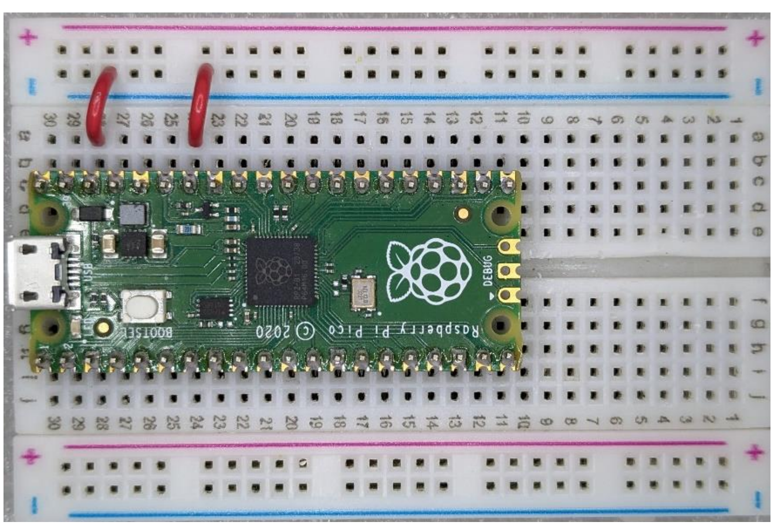
The circuit uses +5VDC from the Pico. It is important to double check wiring and component orientation. Miswiring can place 5V on one of the I/O pins and the Pico will be damaged.



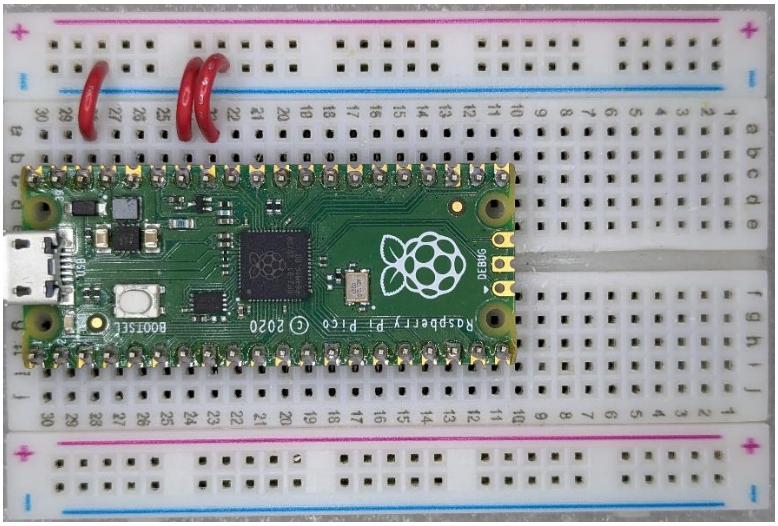
Insert the Pico into the breadboard.



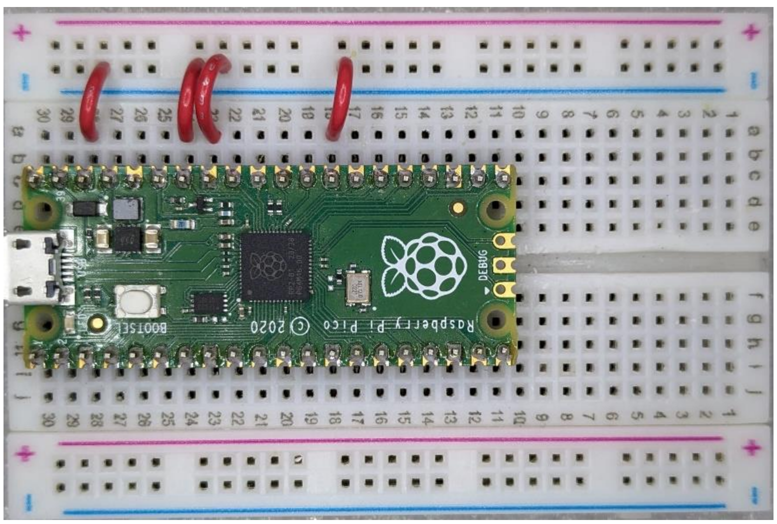
Add a jumper wire from Pico pin 38 to the minus bus.



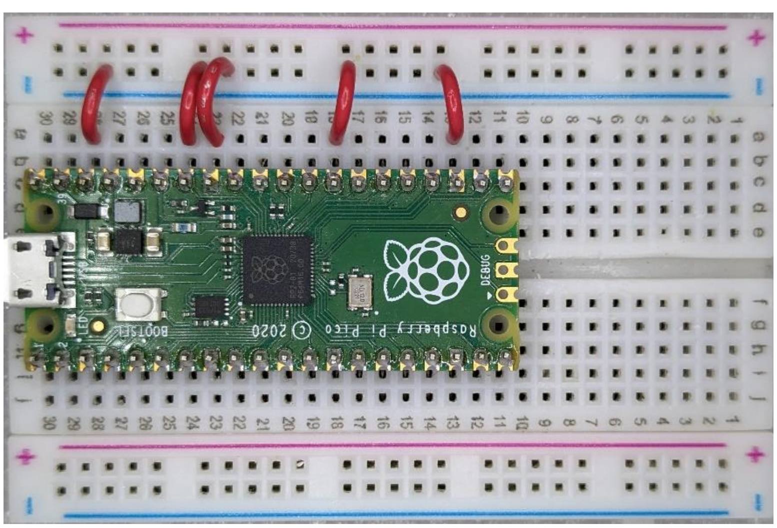
Add a jumper wire from Pico pin 34 to the minus bus.



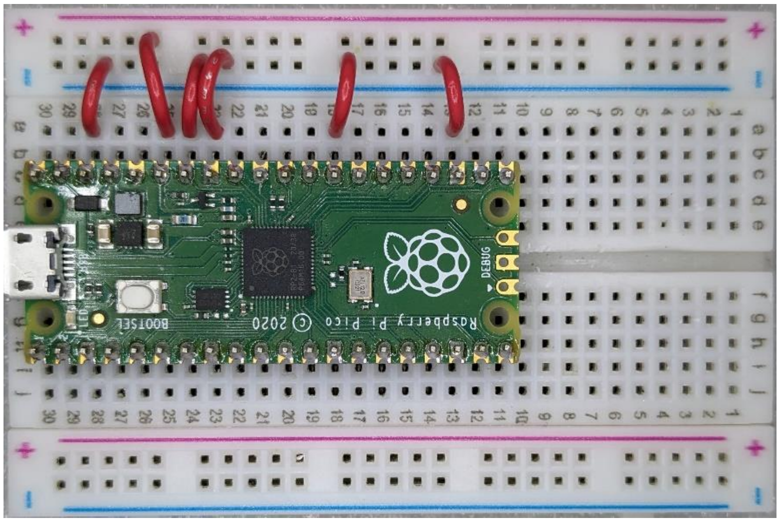
Add a jumper wire from Pico pin 33 to the minus bus.



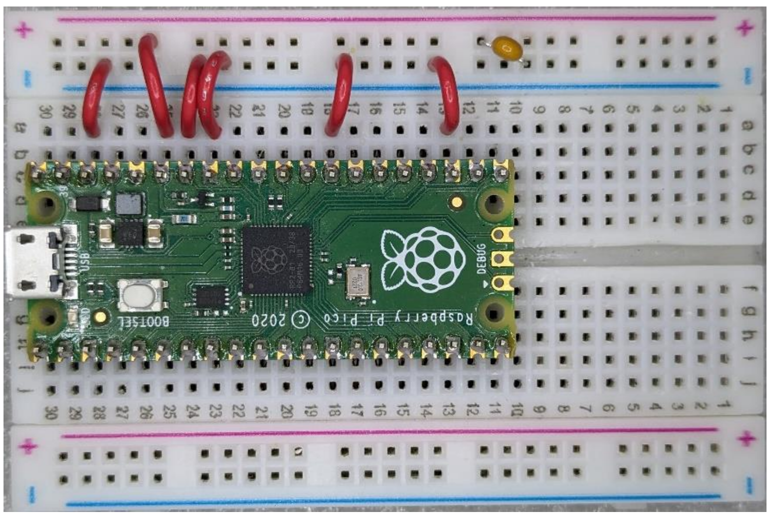
Add a jumper wire from Pico pin 28 to the minus bus.



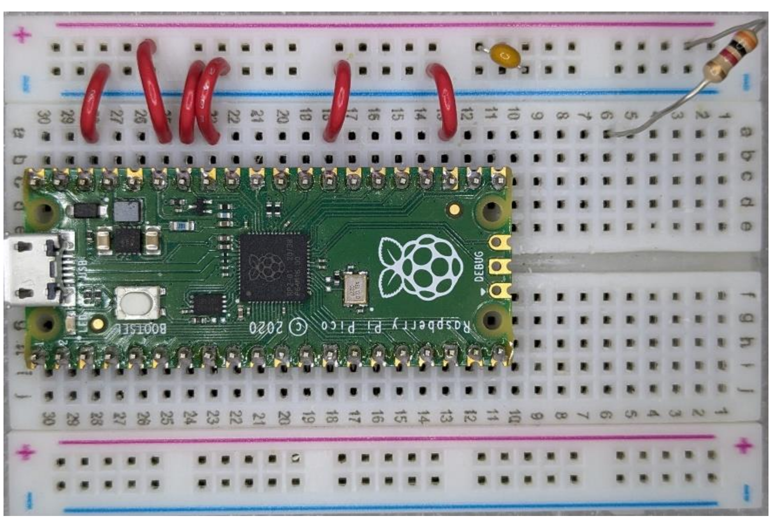
Add a jumper wire from Pico pin 23 to the minus bus.



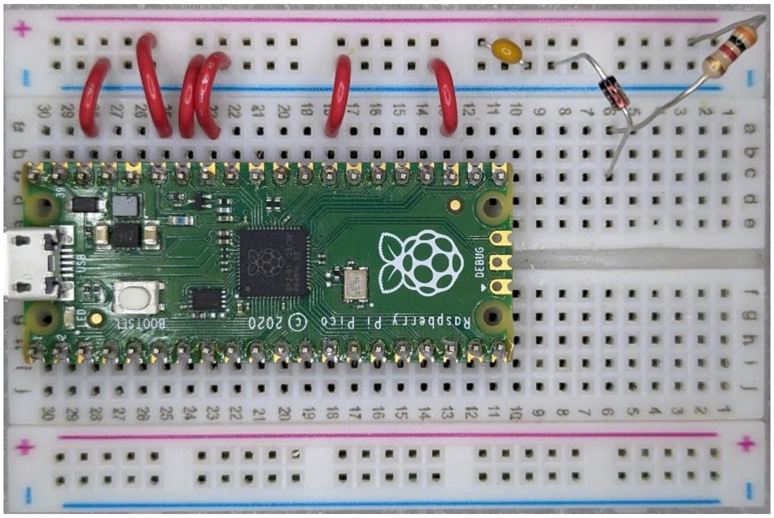
Add a jumper wire from Pico pin 35 to the plus bus.



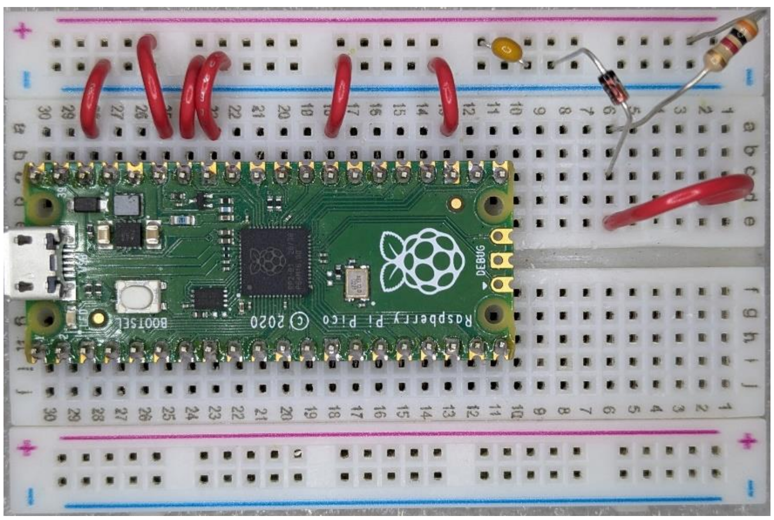
Add a 0.1uF capacitor between the minus bus and the plus bus.



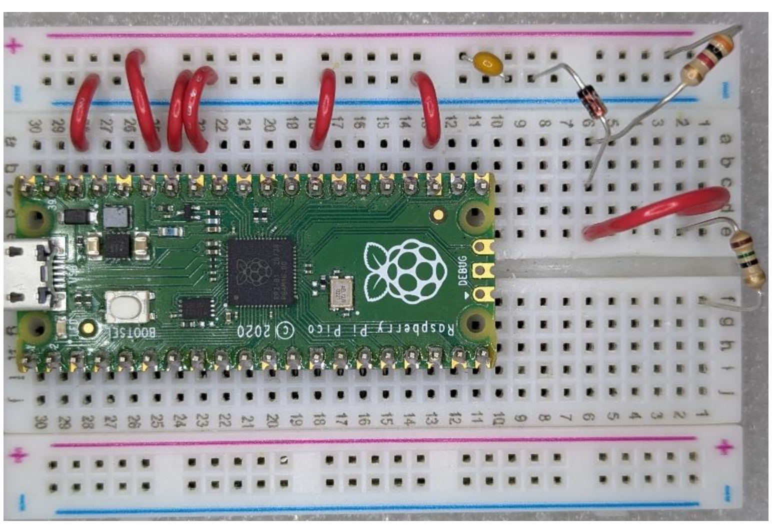
Add a 3K resistor between the plus bus and breadboard bus 6a. All component leads are kept a full length. Trim the ends if there is adhesive from a tape strip.



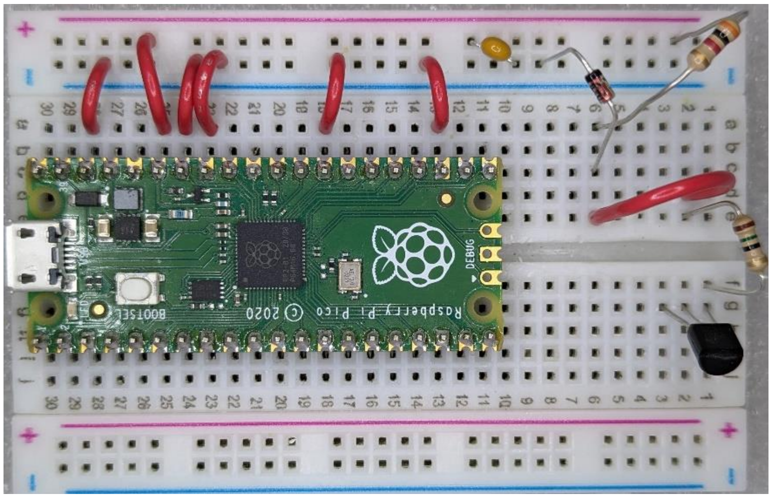
Connect the diode from breadboard bus 6c to the negative bus.



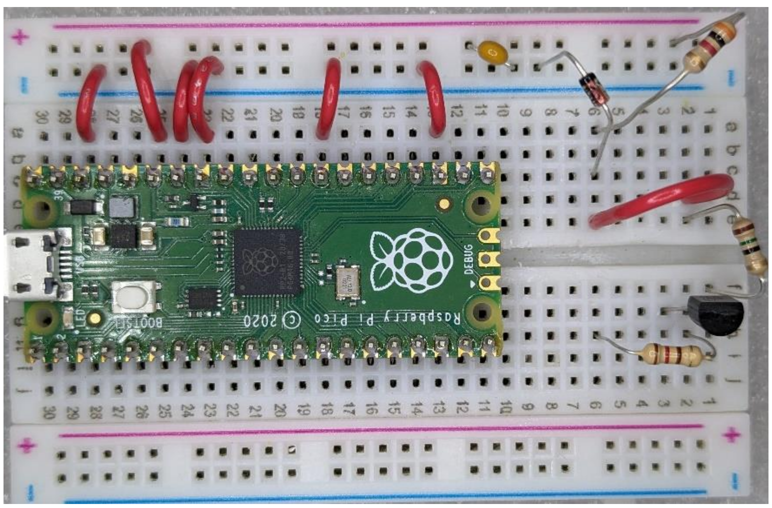
Add a 2” jumper wire from breadboard bus 6e to bus 2d. This wire is used as a thermal isolator from the 15ohm resistor.



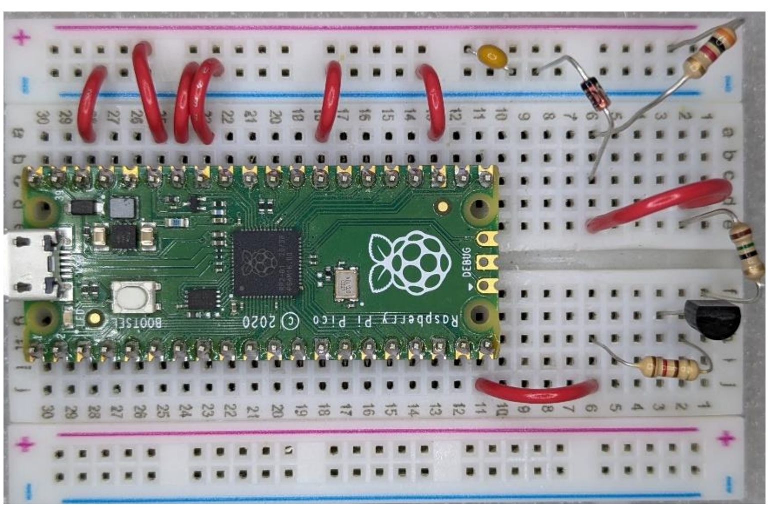
Add a 15 ohm resistor from breadboard bus 2e to bus 1f.



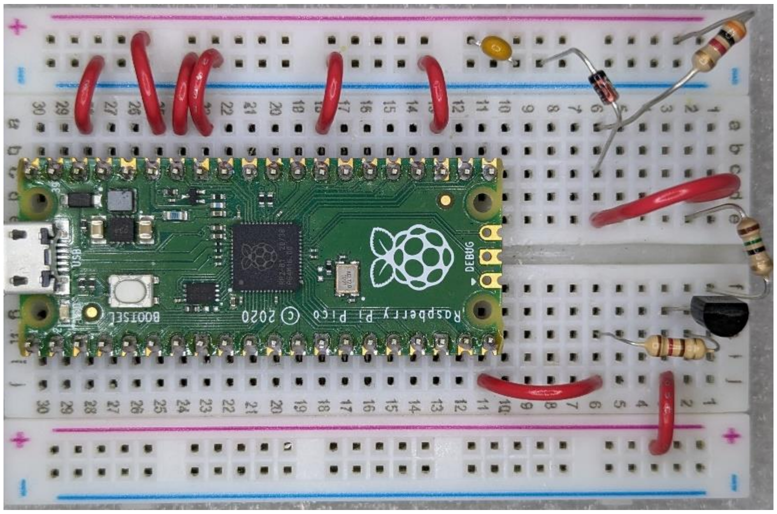
Insert the 2N2222 transistor into the breaboard. Note the flat side points up in the photo. The Emitter connects to bus 1g, the Base to bus 2g and the collector to bus 3g.



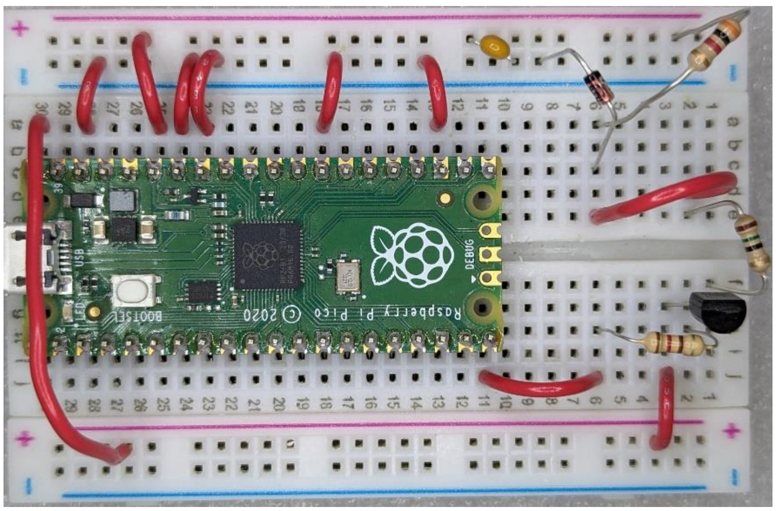
Insert a 120 ohm resistor from the transistor Base at bus 2h to bus 6h.



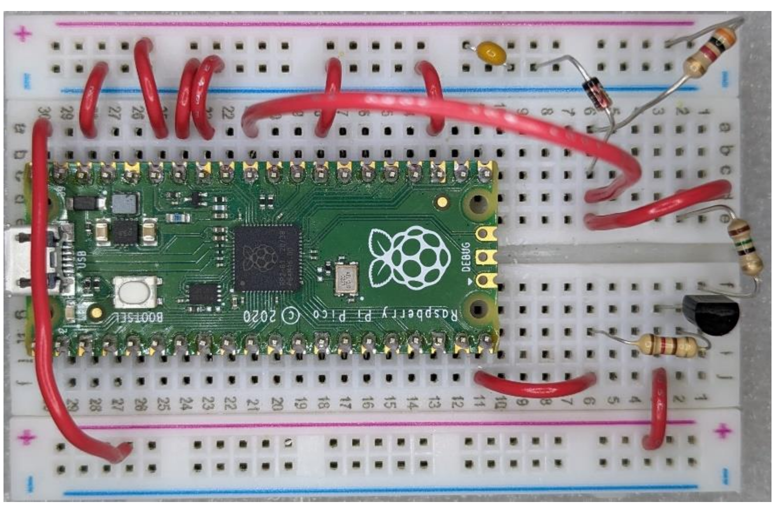
Add a jumper wire from bus pin 6j to Pico pin 20 (D15).



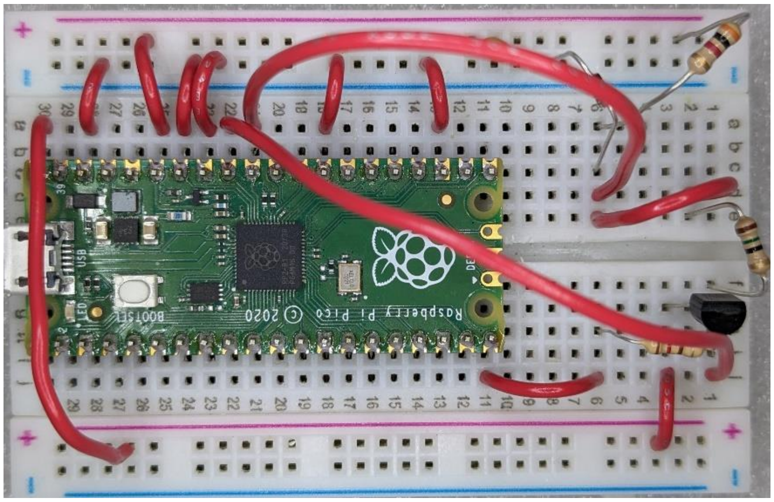
Add a jumper wire from the transistor Collector at bus 3j to the bottom plus bus. Note that the two plus busses are not connected together. The top bus is 3.3V and the bottom one 5V



Add a jumper wire from Pico pin 40 to the bottom plus bus.



Add a jumper wire from Pico pin 21 to the 1N914 anode at bus 6d.



Add a jumper wire from Pico pin 22 to the 2N2222 Emitter at bus 1j.

- Set D15=3.3V turning heating on
  - Heat for 95ms
  - Measure voltages
    - Calculate power
    - mw = 1000 * $(3.3/4096)^2$ * (A0)*(A1-A0) / 15Ω = 4.33e-5 * A0 * (A1-A0)
- Set D15=0V turning heating off
  - Wait to stabilize 5ms
  - Measure A0 diode voltage at 0.9mA
    - Calculate temperature in °C = (3.3/4096)*(ref-A0)/0.002 = 0.403*(ref-A0)

The heating-temperature measurement algorithm is simple. Power is turned on and off. During the longer on-cycle, power is measured. During the short off cycle, temperature is measured. Note that A0 and A1 refer to the analog input channel returned integer values.

- Measure diode voltage at zero power
  - Save voltage as ref2
- Measure diode voltage and power with heating on
  - Save A0 diode reading as ref
  - Save power and diode temperature in array[0]
- Loop 499 test points
  - Heat for 95ms
    - Measure and save power in indexed array mw[]
  - Power off for 5ms
    - Measure and save diode temperature in indexed array
- Turn heating off
- Loop 500 test points
  - Delay 100ms
  - Measure and save power in array
  - Measure and save diode temperature in array using ref2 instead of ref
- Calculate heating power from array values
  - Divide maximum temperature by power to get °C/watt
- Find the cooling time constant
  - Locate array value that is 37% of temperature delta
  - Time constant is the index * 100ms

Putting it together, we simply add a couple of loops to save data during the heating and cooling cycles. The thermal resistance is calculated from the power and temperature data. The thermal time constant is calculated from the cooling temperature data.

The program code is available at https://github.com/simple-circuit/Pico-Thermal-Resistance-Testing in the ***thermal-tester-pico*** folder. You can also drag and drop the ***thermal-tester-pico.uf2*** file into the Pico boot drive. We will go through the code for the educational aspect, where you can enter then save and compile the code.

```
//Pico thermal resistance tester. Simple-Circuit 2023
volatile long analog_avg;    //global variables
volatile long analog_avg2;
volatile float watts[1000];
volatile float deltac[1000];
volatile unsigned long t;
```

Two long variables are used to accumulate ADC averages. Float arrays store diode power in mW and the change in temperature in Celsius. The t variable is used for loop timing.

```
void setup() {
 pinMode(15,OUTPUT);         //pin-15 drives the emitter follower
 digitalWrite(15,0);
 Serial.begin(460800);       //set your serial terminal to 460800 baud
 analogReadResolution(12);   //set the ADC to read 3.3V/4096 = 0.81mV resolution
}
```

Pin 15 controls the power on/off circuit. Although the baud rate is set for 460800, the data is sent at maximum speed and most settings work. The ADC is set for 4096 levels at 12-bits.

```
void readADC(void){
      analogRead(A0);
      analog_avg = 0;
      for (int j = 0; j<64; j++) analog_avg += analogRead(A0);  //average 64 readings to reduce noise
      analog_avg =  analog_avg >> 6;                          //shift to 12-bit value
      analogRead(A1);
      analog_avg2 = 0;
      for (int j = 0; j<8; j++) analog_avg2 += analogRead(A1);  //average 8 readings to reduce noise
      analog_avg2 =  analog_avg2 >> 3;
}
```

Analog channel A0 is averaged for 64 readings. This is the temperature measurement value which has a resolution of about 0.8mV. Channel A1 is averaged for eight readings. It is used to calculate power and a higher noise is tolerable. Global variable analog_avg contains the A0 value and analog_avg2 contains the A1 value.

```
void loop() {
  char c;
  int i;
  float mw, ref, ref2, degc;
   c=' ';
   digitalWrite(15,0);                 //turn heating off
   while (c!='r'){                     //start the test when an 'r' is typed
     while (Serial.available() == 0);
     c = Serial.read();
   }
```

The first function of the main loop is to wait to start. Reception of a lower case ‘r’ begins the test.

```
Serial.println("mw, delta_C");
 readADC();
 ref2 = 0.403*analog_avg;       //diode cool reference
 digitalWrite(15,1);            //start heating
 delay(100);                    //heat for 100ms
 readADC();                     //measure the heating power
 mw = analog_avg*(analog_avg2-analog_avg)*0.0000433;
 digitalWrite(15,0);           //drop the heating
 delay(5);                     //wait for system to settle
 readADC();                    //measure the diode voltage
 t = millis() + 100;           //initialize loop timer
 digitalWrite(15,1);           //turn on heating
 ref = analog_avg*0.403;       //save the diode measure starting point
 degc = ref - analog_avg*0.403;
 watts[0] = mw;                //log the first values
 deltac[0] = degc;
 Serial.print(mw);             //print the data
 Serial.print(", ");
 Serial.println(degc);
```

Header text for power and temperature allow the plotter to designate the plotted values. An initial cold diode temperature is recorded in ref2. A single heating pulse and temperature measurement record the initial heating cycle start in ref. The heating cycle causes a bias in the measurement and two different zero references are needed.

```
for (i=1; i<500; i++){
 while (millis() < t);         //sample every 100ms
 t = t + 100;
 readADC();                    //measure heating power
 mw = analog_avg*(analog_avg2-analog_avg)*0.0000433;
 watts[i] = mw;
 digitalWrite(15,0);           //drop the power and mesure diode voltage
 delay(5);
 readADC();
 digitalWrite(15,1);          //turn heat on
 degc = ref - analog_avg*0.403;
 deltac[i] = degc;
 watts[i] = mw;              //log the data
 Serial.print(mw);           //print the data
 Serial.print(", ");
 Serial.println(degc);
}
```

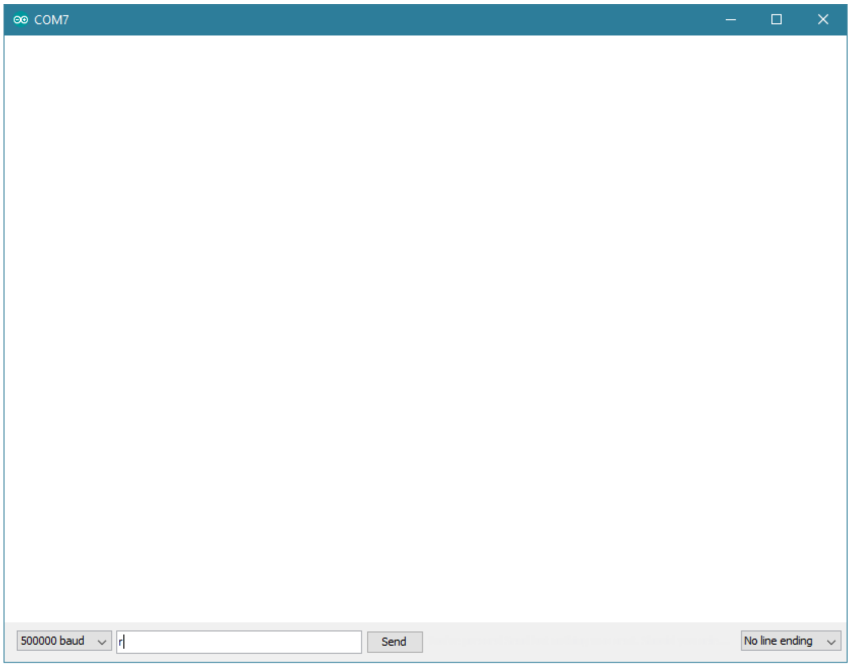
The heating cycle is a loop of 95ms power on, 5ms delay, then reading the temperature. The zero starting point is subtracted from the temperature to generate the delta. The subtraction is reversed to correct for the negative slope.

```
 digitalWrite(15,0);          //all done, turn of heating

for (i=500; i<1000; i++){
 while (millis() < t);
 t = t + 100;
 readADC();                  //read the diode voltage as it cools
 mw = analog_avg*(analog_avg2-analog_avg)*0.000065;
 readADC();
 degc = ref2 - analog_avg*0.403;
 deltac[i] = degc;           //log the data
 watts[i] = mw;
 Serial.print(mw);           //print the data
 Serial.print(", ");
 Serial.println(degc);
}
```

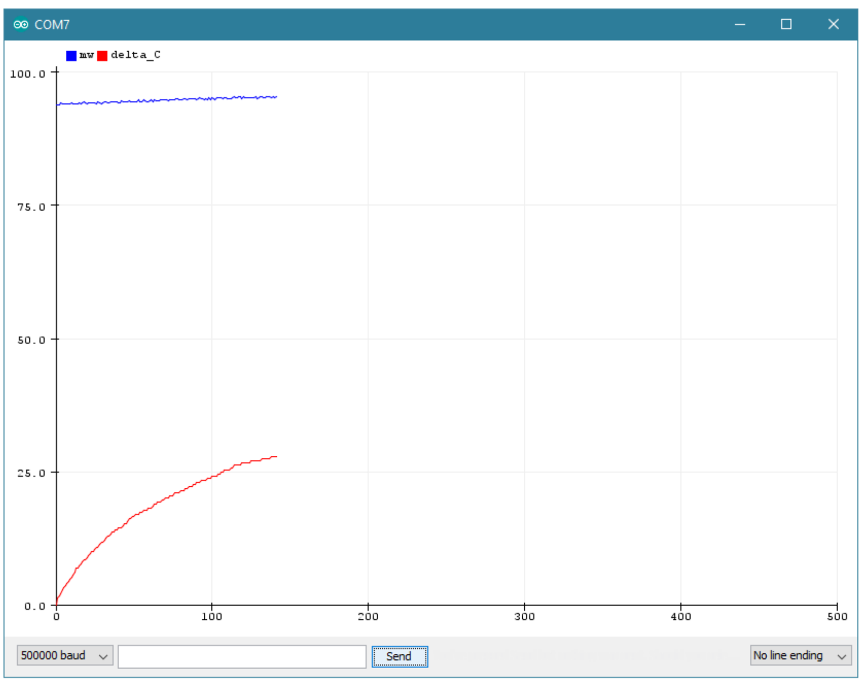
The cooling cycle is the same as the heating cycle except no power is applied. The initial cold temperature ref2 is used to calculate the delta. With no power applied, the measurement bias is not present.

```
 mw = 0.0;                     //calculate the average power
for (i=1; i<500; i++) mw = mw + watts[i];
mw = mw / 525.26;
degc = 1000.0 * deltac[499] / mw;   //calculate the thermal resistance
Serial.print(degc,0);               //print it
Serial.print("C/W, ");
mw = (deltac[501] - deltac[999])*0.37 + deltac[999];


for (i=501 ; i<999; i++){           //find the thermal time constant
 if (deltac[i]<=mw){
   Serial.print((i-501)*0.100,1);   //print it
   Serial.println("sec_TC");
   break;
 }
}

}
```

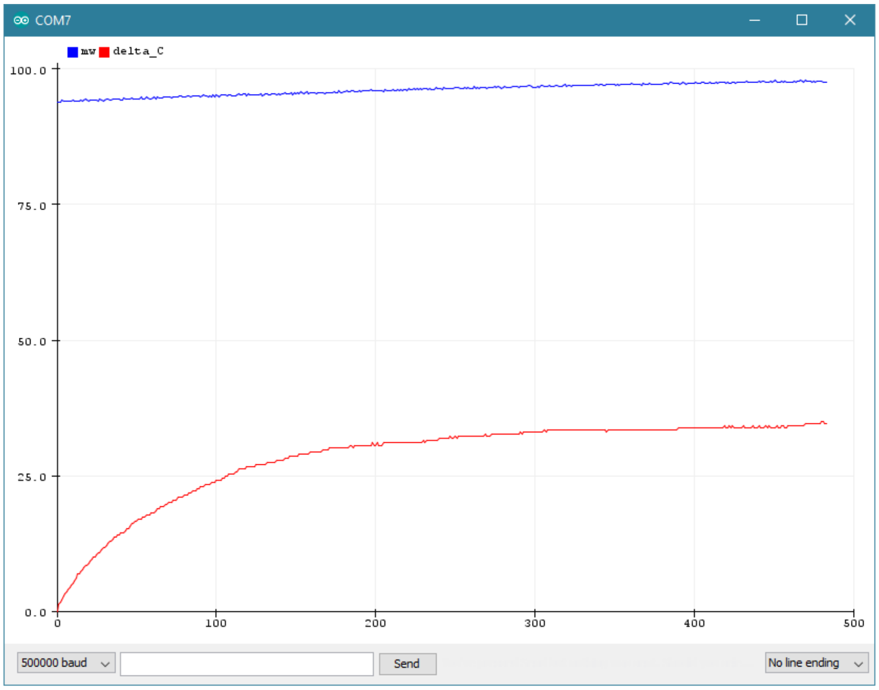
Finally, power readings are averaged and compensated for the 95% duty cycle. The thermal time constant is measured by locating the cooling delta that has dropped to 63% of the full change.



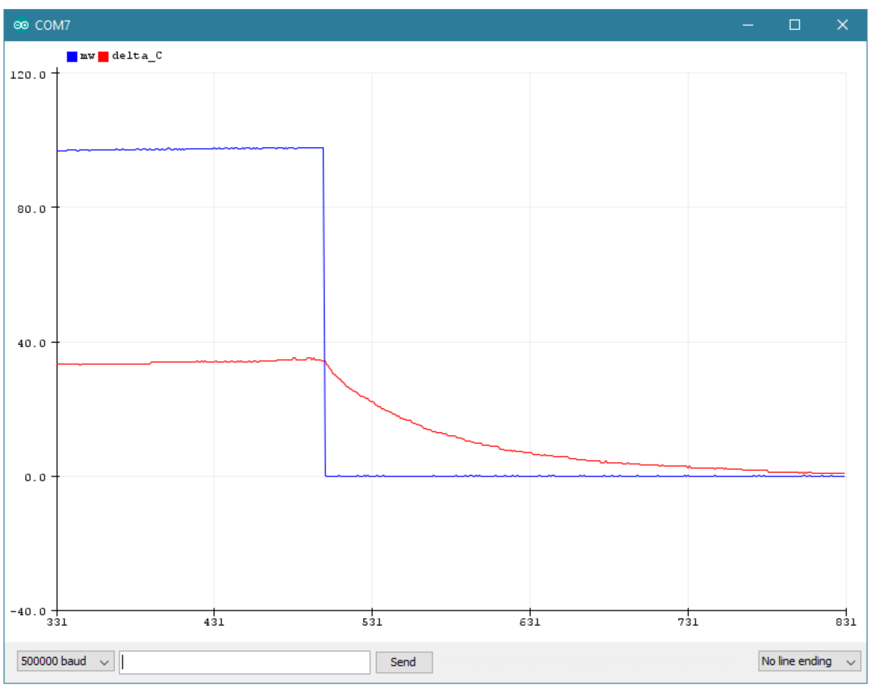
Save your code then compile and upload it to the Pico. Start the Serial Plotter under Tools. Send an ‘r’ character to start the test.



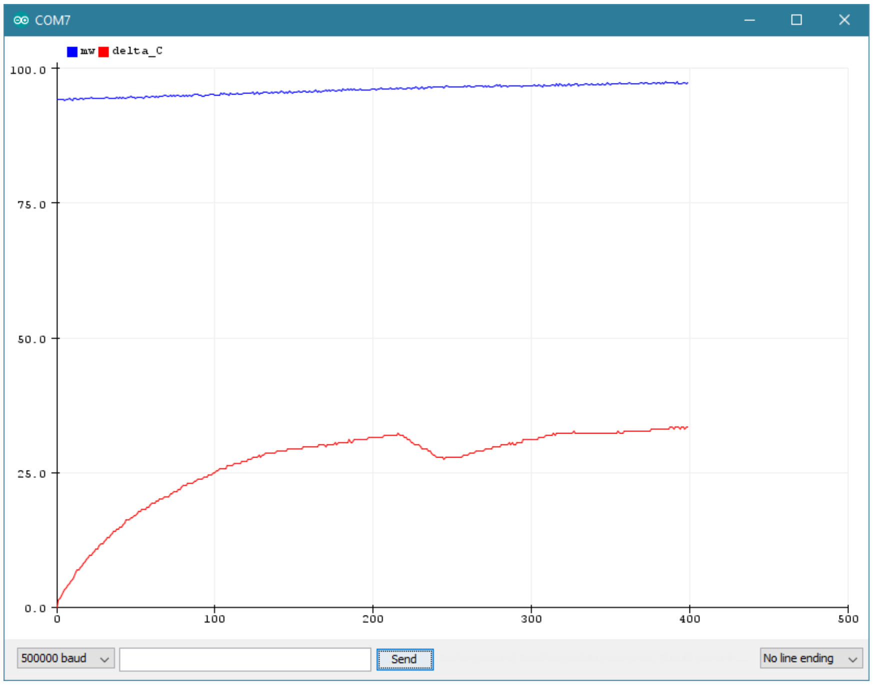
The diode will start heating. The power in mW is shown in the blue plot and the delta temperature in degrees C is shown in the red plot.



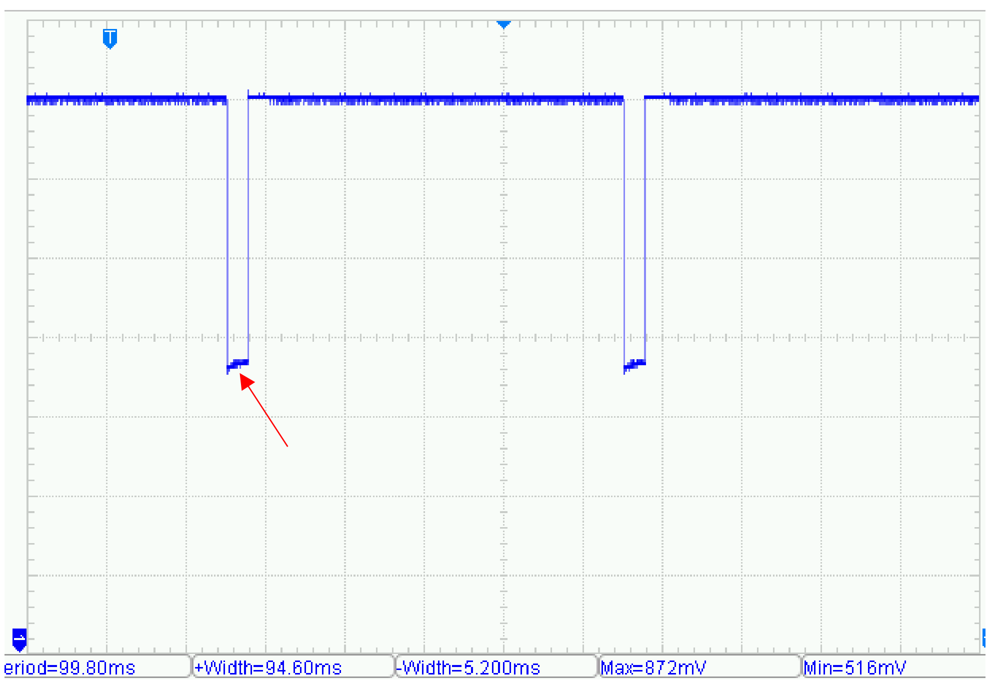
The heating cycle runs for 50 seconds to ensure the die temperature stabilizes.



After the heating cycle, the exponential cooling is recorded.



Diode heating and cooling is influenced by air currents. Even breathing on the diode can change its temperature.



Testing the program and observing the diode voltage shows power on and off cycles. Note the small recovery of the off cycle. This recovery was also observed using only resistors instead of the diode. It is believed to be a recovery of Pico electronics after the large voltage step. This is the reason for the 5ms delay before taking the ADC sample.

- Take away:
  - Diodes allow measurement of temperature through the forward voltage change of
    - -2mV/°C.
  - Self-heating produces a power flow that increases the die temperature.
  - Temperature change from self-heating is how thermal resistance is measured.
  - Most silicon devices have a diode structure accessible from external connections. These diodes can be used to measure temperature. However, self-heating may require a more complex switching circuit.