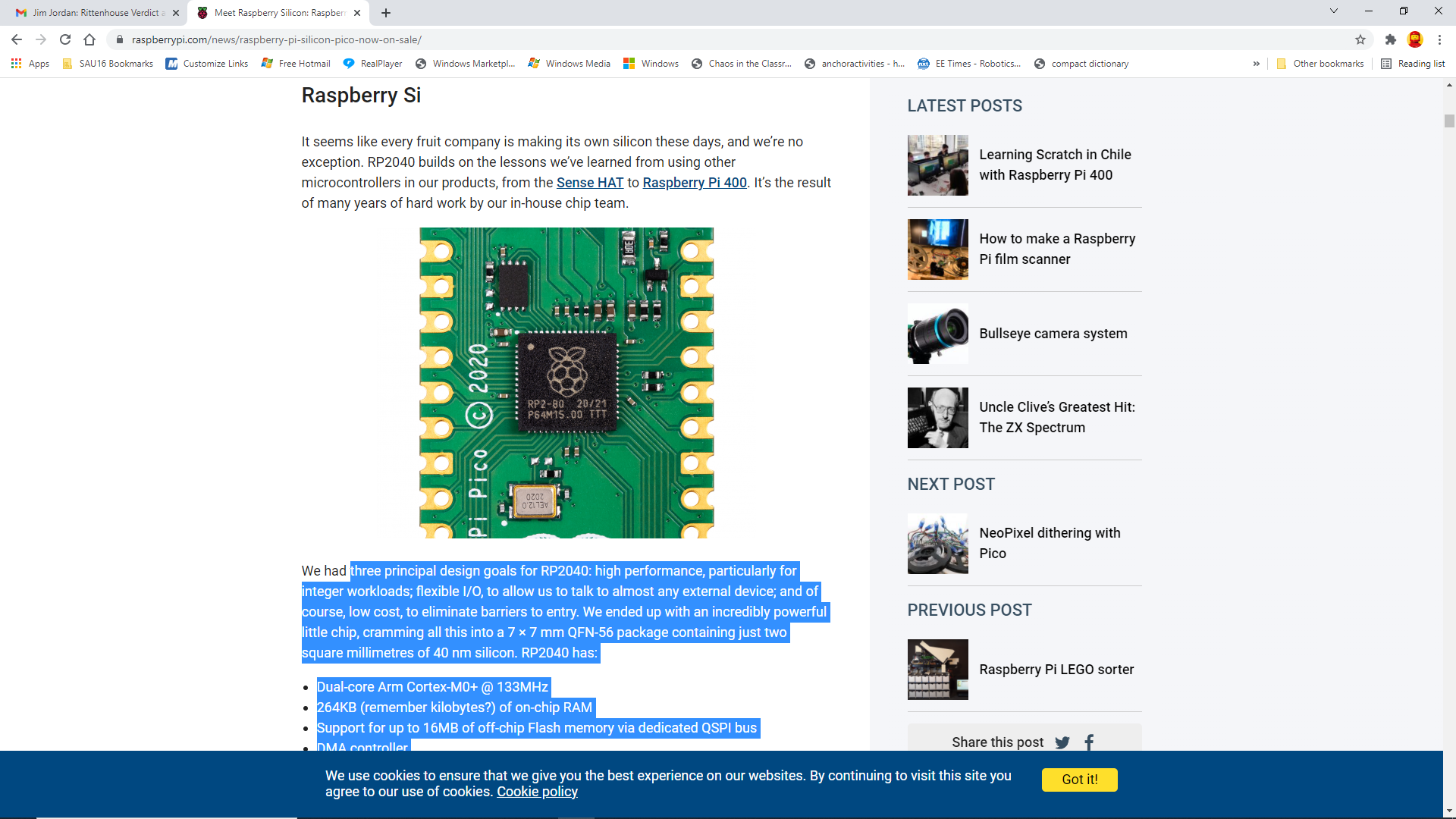
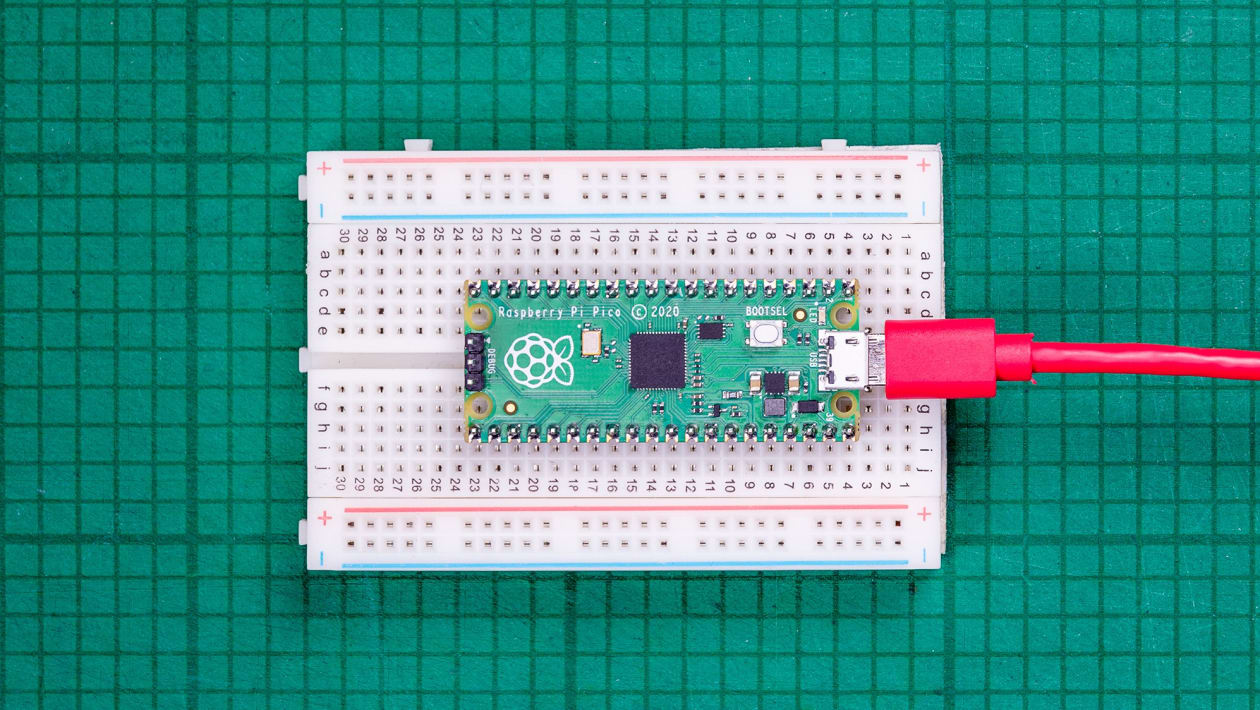
**Create a Morse Code Light with Raspberry Pi Pico**

The Raspberry Pi Pico is an inexpensive ($4) single board computer based on the RP2040 Processor.

There were three principal design goals for RP2040: high performance, particularly for integer workloads; flexible I/O, to allow us to talk to almost any external device; and of course, low cost, to eliminate barriers to entry. We ended up with an incredibly powerful little chip, cramming all this into a 7 × 7 mm QFN-56 package containing just two square millimeters of 40 nm silicon. RP2040 has:

* Dual-core Arm Cortex-M0+ @ 133MHz
* 264KB (remember kilobytes?) of on-chip RAM
* Support for up to 16MB of off-chip Flash memory via dedicated QSPI bus
* DMA controller
* Interpolator and integer divider peripherals
* 30 GPIO pins, 4 of which can be used as analogue inputs
* 2 × UARTs, 2 × SPI controllers, and 2 × I2C controllers
* 16 × PWM channels
* 1 × USB 1.1 controller and PHY, with host and device support
* 8 × Raspberry Pi Programmable I/O (PIO) state machines
* USB mass-storage boot mode with UF2 support, for drag-and-drop programming

The Pico can easily be programmed using MicroPython (a subset of Python) and the Thonny Integrated Development Environment. You can connect it to your computer using a USB cable with a microUSB connector. Our Pico’s are mounted to a 2000 contact breadboard to allow easy connection for more sophisticated physical computing projects.

Our Pico’s already have MicroPython installed on them.

**Source Code**

# morse.py

# This program will convert a user provided phrase into Morse Code

# and flash an LED with the code

# Author: nmessa

# Date: 10.27.2021

from machine import Pin

import time

def codeGen(s):

output = ''

s = s.lower() #put string in lower case to reduce dictionary size

for i in range(len(s)):

if s[i] == '\n':

output += '\n'

continue

output += convert(s[i])

output += ' '

print (output)

for i in range(len(output)):

if output[i] == '.':

dot()

if output[i] == '-':

dash()

if output[i] == ' ':

time.sleep(0.1)

def convert(letter):

#International Morse Code dictionary

table = {'a':'.-', 'b':'-...', 'c':'-.-.', 'd':'-..', 'e':'.',

'f':'..-.', 'g':'--.', 'h':'....', 'i':'..', 'j':'.---',

'k':'-.-', 'l':'.-..', 'm':'--', 'n':'-.', 'o':'---',

'p':'.--.', 'q':'--.-', 'r':'.-.', 's':'...', 't':'-',

'u':'..-', 'v':'...-', 'w':'.--', 'x':'-..-', 'y':'-.--',

'z':'--..', ' ':' ', '1':'.----', '2':'..---',

'3':'...--', '4':'....-', '5':'.....', '6':'-....',

'7':'--...', '8':'---..', '9':'----.', '0':'-----'}

if letter in table.keys(): #check that letter is in dictionary

return table[letter]

else: #handle other characters where code does not exist in dictionary

return '' # 2 single quotes

def dot():

led.toggle()

time.sleep(0.1)

led.toggle()

time.sleep(0.1)

def dash():

led.toggle()

time.sleep(1)

led.toggle()

time.sleep(0.1)

#Main program

while True:

led = Pin(25, Pin.OUT)

phrase = input("Enter a phrase: ")

codeGen(phrase)