Introduction to Physical Computing with the Pi Pico and Python

Bill Ball, bill@tinkerfarm.net, www.tinkerfarm.net

Agenda:

A. Why the Pi Pico?

Single board computers vs. microcontrollers: the Raspberry Pi Pico is not a Raspberry Pi

Capabilities and limitations of the basic Pico

B. Parts of the Ecosystem

Thonny

Micropython

Pico

Breadboarding

C. Working with Example Scripts and Circuits

1_hello.py Thonny IDE, micropython and REPL.

2_blink.py Modules, GPIO pins, loops.

3_button.py Breadboarding circuits, GPIO input.

4_buzzer_threads.py Functions, threading.

5_pir_interrupts.py Sensors, interrupts.

6_pot_pwm.py Potentiometers, ADC, PWM.

7_I2C_display.py Custom modules, I2C, display output

8_PIO_neopixels.py PIO, neopixels

D. The Next Level

Power considerations

Onboard storage, programs and data

Additional communications protocols: I2C, SPI, Serial, I2S via PIO

Other boards and embedded applications, SDKs

E. Resources

At the makerspace: electronics area, other classes, meetup, one-on-one

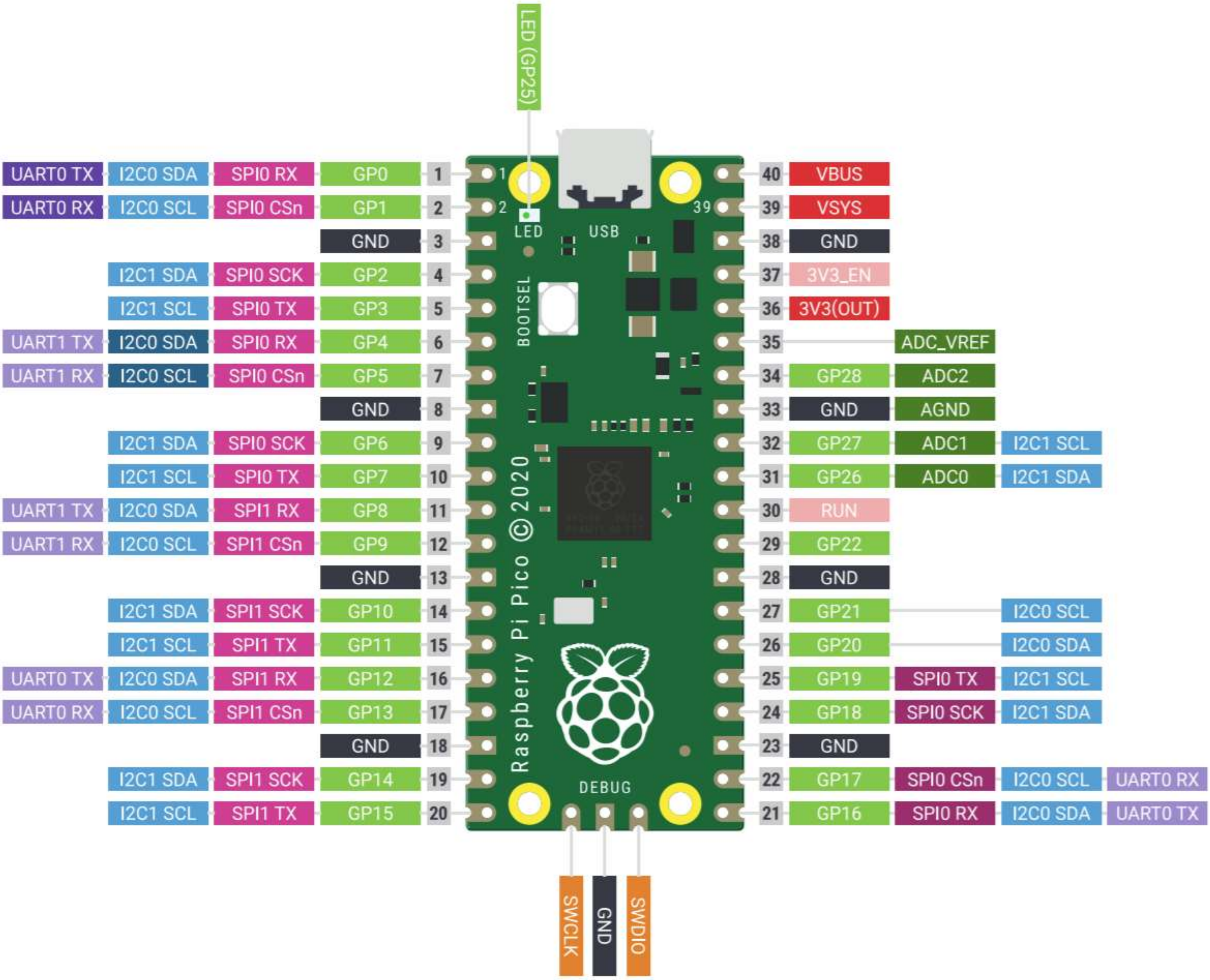
Raspberry Pi Foundation:
https://www.raspberrypi.org/products/raspberry-pi-pico/

Boards, components, tutorials:

Adafruit: https://www.adafruit.com/
Sparkfun: https://www.sparkfun.com/

Seeed Studio: https://www.seeedstudio.com/
Amazon, Digikey, Microcenter, Tinkersphere, Aliexpress

F. Pi Pico pinout & key specs



Dual-core Arm Cortex-M0+ processor, flexible clock running up to 133 MHz
264KB on-chip SRAM
2MB on-board QSPI Flash
26 multifunction GPIO pins, including 3 analogue inputs
2 × UART, 2 × SPI controllers, 2 × I2C controllers, 16 × PWM channels
1 × USB 1.1 controller and PHY, with host and device support
8 × Programmable I/O (PIO) state machines for custom peripheral support
Supported input power 1.8–5.5V DC