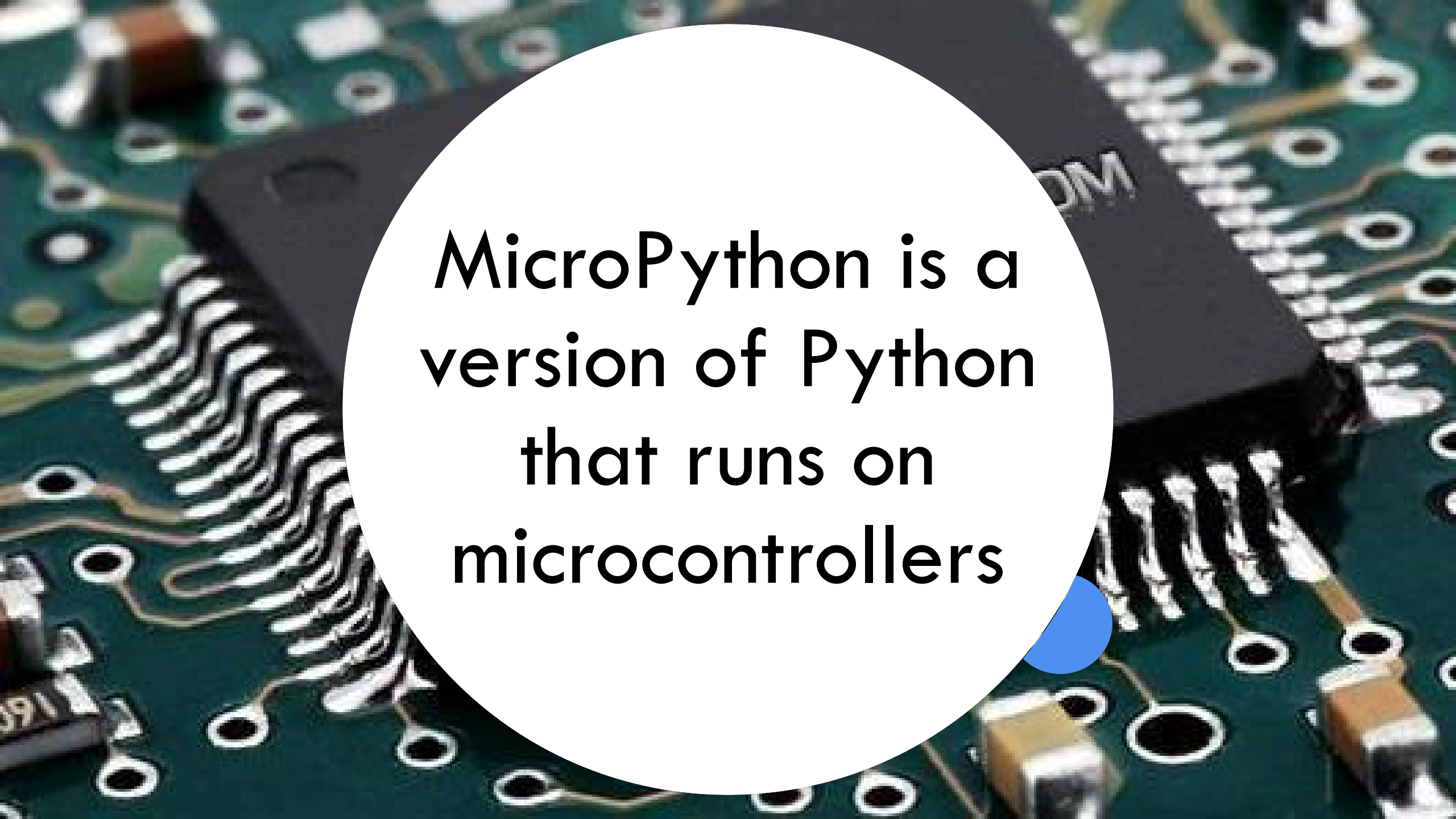
Integrating Computation with MicroPython and Sensors

John Liu
Saint Cloud State University

Agenda

- What is MicroPython?
- Platforms running MicroPython
- Simple examples



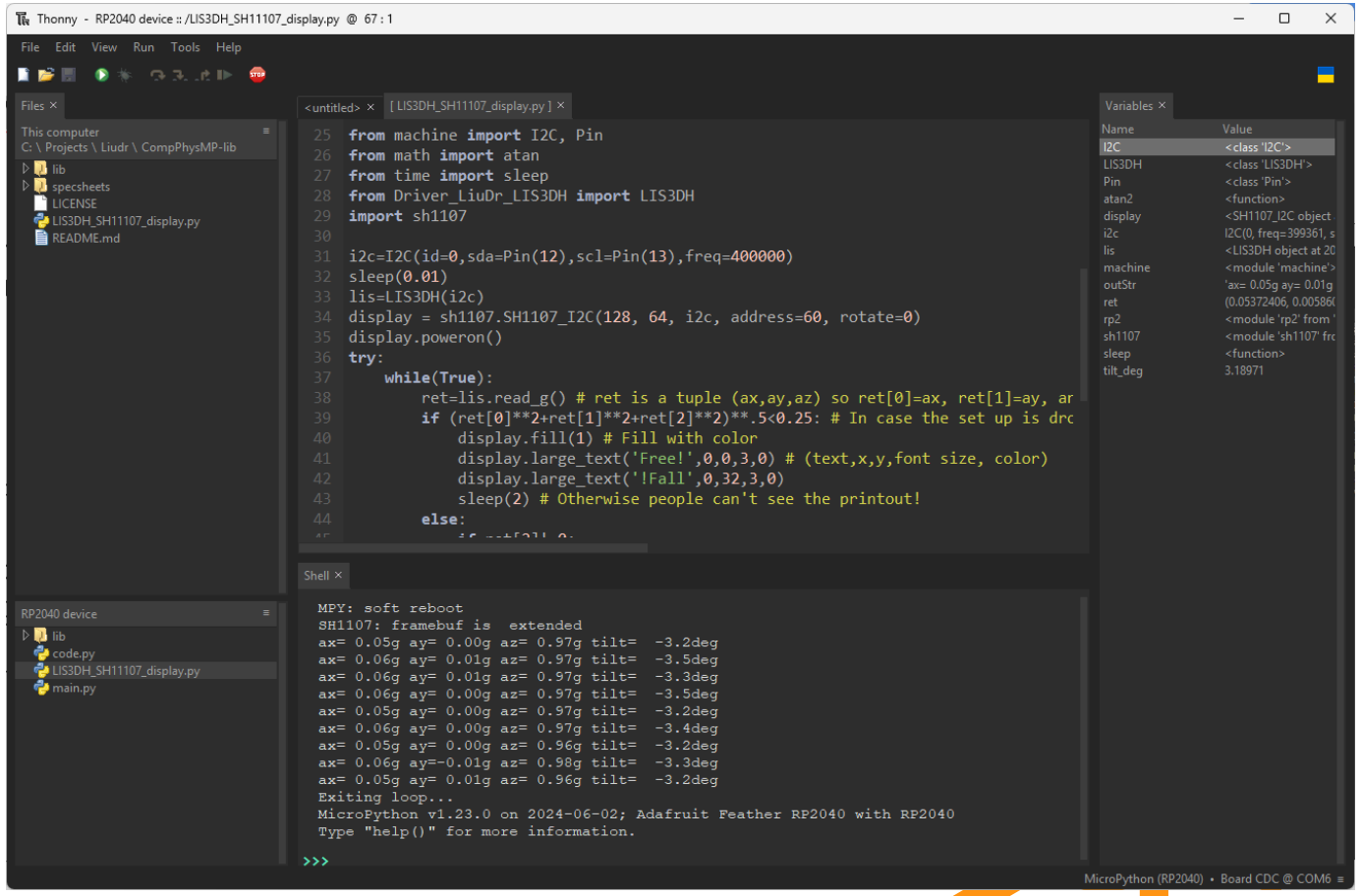
MicroPython is a version of Python that runs on microcontrollers

Why MicroPython?

- Same ease-of-use as PC Python, but runs directly on a microcontroller
- Interact directly with sensors
- Control motors, lights, displays etc.
- Construct a lab or demo that works without constant need for IT support
- Perfect for hands-on learning experience when integrating computation in labs

Thonny IDE

- Syntax highlighting
- File browser and transfer between PC and board
- Execute code on PC or board
- Shell for REPL
- Variables list for debugging
- More features

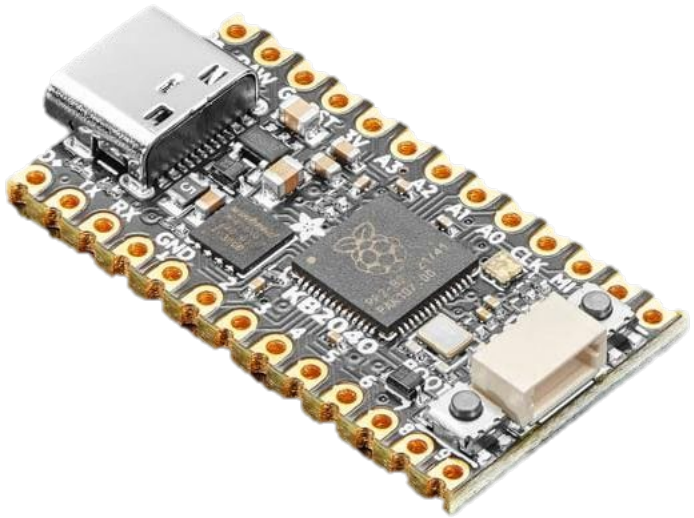


Hardware platforms running MicroPython

Many and more each year

Hardware

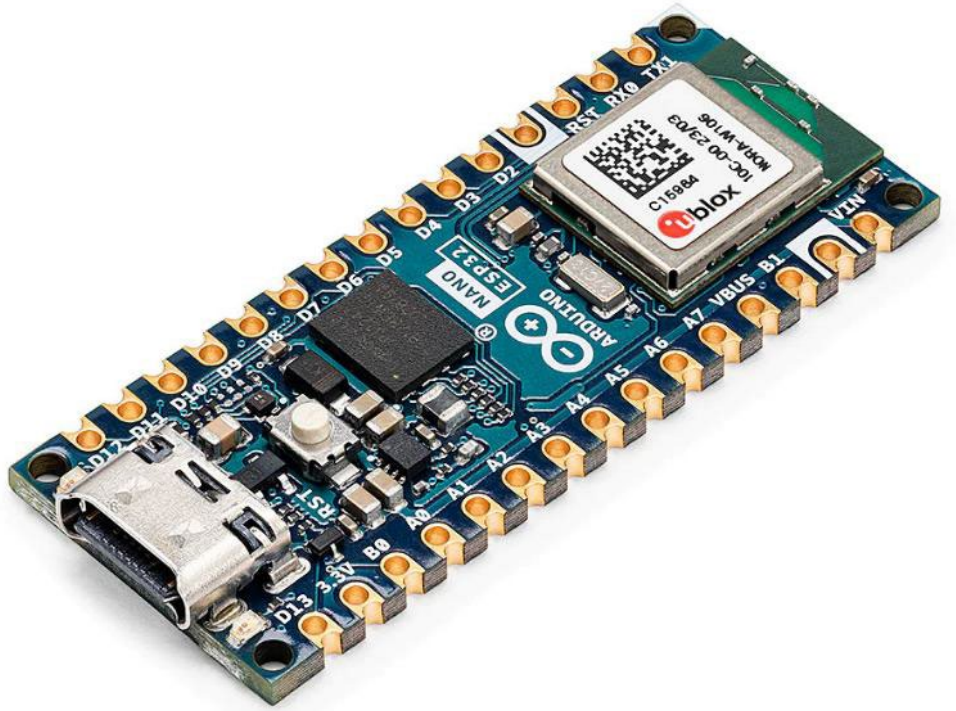
- Raspberry Pi Pico/Pico W/Pico 2-based boards
- ESP32/ESP32-S3/ESP32-Pico-based boards
- IMXRT-based boards
- Other boards or platforms such as Lego SPIKE, Vex IQ V2, based on nRF52840, ATSAMD21, ATSAMD51, STM32 etc.
- 100+MHz processor
- 256KB+ RAM (variables), 2MB+ FLASH (code)



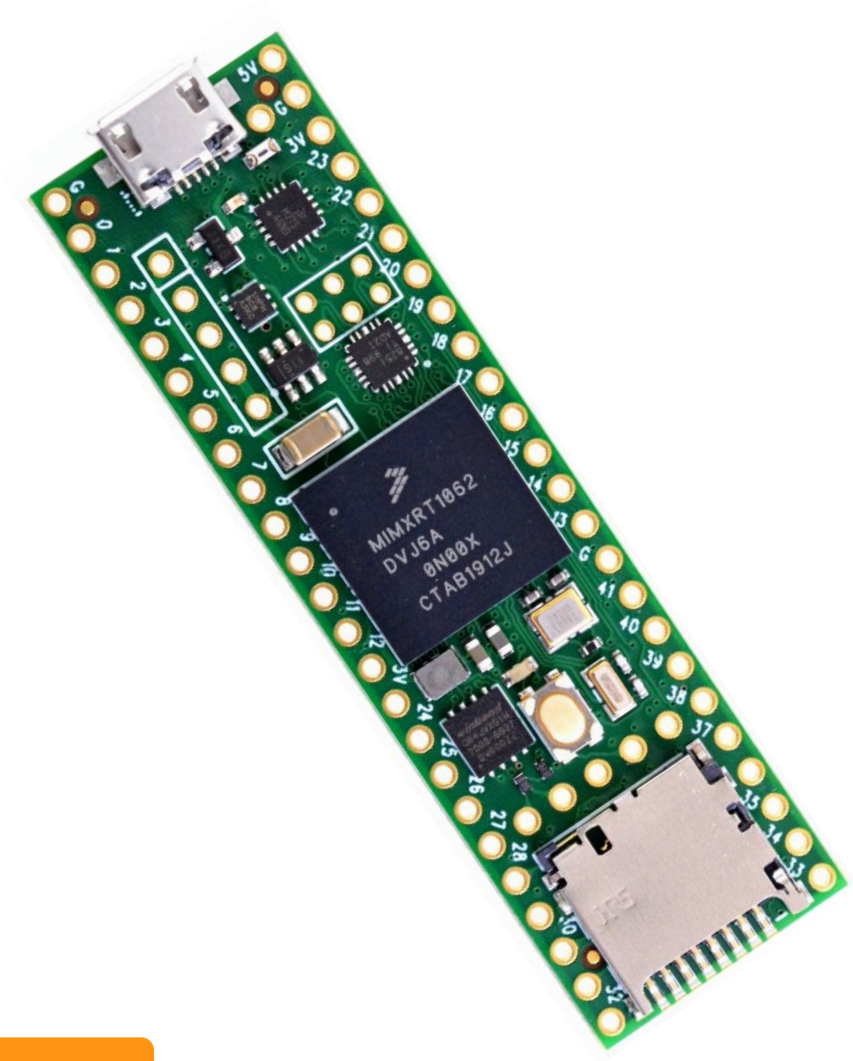
RP2040
Dual 133MHz CPU
260KB RAM
2-8MB FLASH

- Adafruit KB2040 260KB/8MB
- Adafruit QT Pi 2040 260KB/8MB
- Adafruit Feather 2040 260KB/8MB
- Raspberry Pi Pico 260KB/2MB
- Raspberry Pi Pico W 260KB/2MB+WiFi

ESP32
Dual 240MHz CPU
520KB RAM
0-8MB PSRAM
4-16MB FLASH
WiFi/Bluetooth

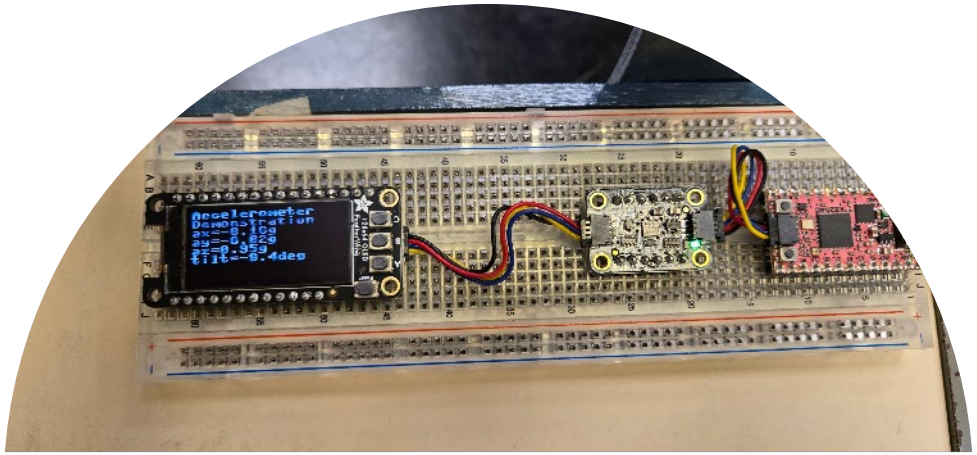


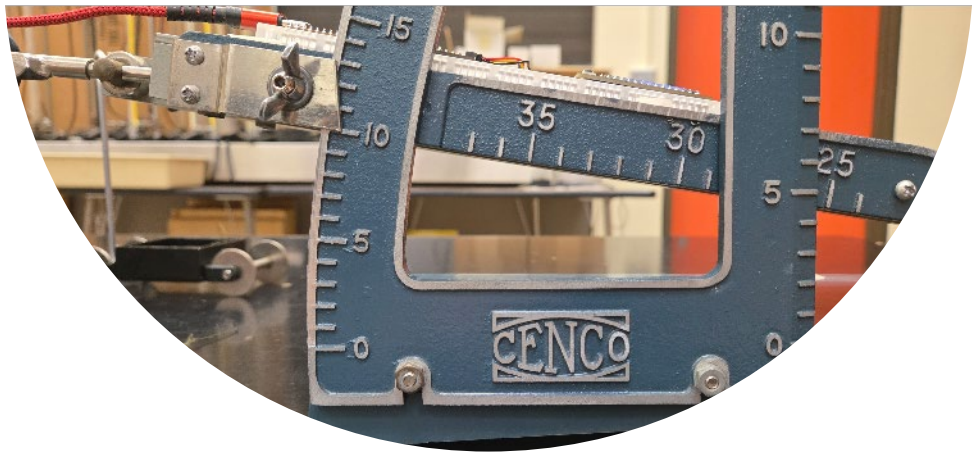
- Adafruit Feather ESP32-V2 2MB/8MB
- Adafruit Qt Pi ESP32 Pico 2MB/8MB
- Arduino Nano ESP32 8MB/16MB
- Espressif ESP32 PICO-KIT-1 520KB/4MB
- TinyPICO 4MB/4MB



IMXRT
500-600MHz CPU
128KB-1MB RAM
2-8MB FLASH

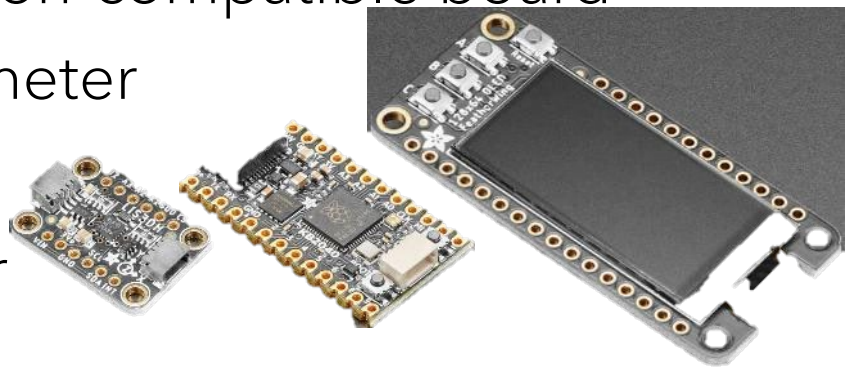
- Adafruit Metro M7 500MHz/28KB/8MB
- PJRC Teensy 4.0 600MHz/1MB/2MB
- PJRC Teensy 4.1 600MHz/1MB/8MB





Making a digital protractor with an accelerometer

- MicroPython-compatible board
- Accelerometer
- Display
- Computer



Code

```
from math import atan2
from time import sleep
from machine import I2C, Pin
from Driver_LiuDr_LIS3DH import LIS3DH
import sh1107
i2c=I2C(id=0,sda=Pin(12),scl=Pin(13),freq=400000)
sleep(0.01)
lis=LIS3DH(i2c)
display = sh1107.SH1107_I2C(128, 64, i2c, address=60, rotate=0)
display.poweron()
while(True):
    ret=lis.read_g() # ret is a tuple (ax,ay,az)
    tilt_deg=atan2(ret[0],ret[2])*57.3 # Calculate tilt angle
    outStr=f'ax={ret[0]:5.2f}g ay={ret[1]:5.2f}g az={ret[2]:5.2f}g tilt={-tilt_deg :6.1f}deg'
    print(outStr)
    display.fill(0)
    display.large_text('{:3d}'.format(-int(tilt_deg+0.5)),0,8,4,1)
    display.large_text('Deg',40,48,2)
    display.show()
    sleep(0.1)
```

Thank you

John Liu

St. Cloud State University

Dept. Mathematics and Physics

zliu@stcloudstate.edu

Code and resources:

https://github.com/opensourcephysicslab/CompPhysMP-lib