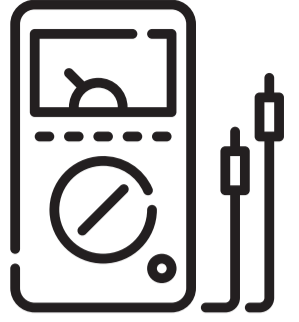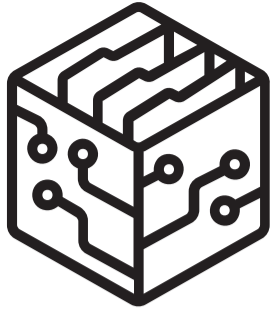# EMBEDDED SYSTEMS

## Skill Tree: Color in the Boxes

Color in the boxes of anything you've already completed, visualize your skills and identify your skill gaps. Get inspired to try new things and tailor the skill tree to suit your own journey by swapping in your own goals.

**ADVANCED**

**BASICS**

(set your own goal)

(set your own goal)

(set your own goal)

(set your own goal)

(set your own goal)

(set your own goal)

- Create a CI/CD System
- Write an Arduino Library
- Write a datasheet
- Edit binary / hex to fix a problem after release
- Design a low current circuit to monitor lithium battery usage
- Use trace debugging to troubleshoot a problem
- Create a software release process
- Fix 49-day reboot problem with a calendar
- Add Machine Learning (ML) to a device
- Teach a class on embedded systems
- Make a dashboard for monitoring remote units
- Write software for programming and testing many boards in manufacturing
- Bit bang a communication driver
- Write custom BLE GATT to communicate between devices
- Use a multi core processor
- Create your own custom PCB
- Use a .map file to debug a crash
- Adapt RTOS to a new board
- Optimize assembly code
- Use multiple architectures in a software image
- Blow fuses to test security in a project
- Modify a Linker File
- Debug a project with a logic analyzer
- Reverse engineer a communication protocol
- Run unit tests in an algorithm sandbox
- Identify a burden voltage issue
- Use Direct Memory Access (DMA)
- Debug a hard fault on a microcontroller
- Write manufacturing test software
- Debug a project with an oscilloscope
- Use a camera in a project
- Bring up a custom board for the first time
- Check a .lst or .map file to troubleshoot a problem
- Use an IMU sensor to track motion in multiple axes
- Update firmware remotely using secure encryption
- Set up a cross compiler
- Use a mutex to safeguard a resource
- Create a circular buffer that uses more than 2 pointers
- Implement MCU sleep in a low power project
- Create a custom USB HID Device
- Use closed loop control (PID) to regulate a project
- Implement an interrupt service routine (ISR)
- Analyse a signal with Fast Fourier Transform (FFT)
- Update firmware remotely (OTA, DFU, FWUP)
- Display text on a screen in a different font
- Use a Real Time Operating System (RTOS)
- Use a step-through debugger to troubleshoot a problem
- Use pulse width modulation (PWM) to control RGB LEDs
- Move a motor
- Erase and rewrite FLASH memory
- Understand how to use a circular buffer
- Configure a timer
- Use Bluetooth Low Energy (BLE) to show a sensor value
- Use Pulse Width Modulation (PWM)
- Implement a state machine
- Analyse a signal with Fast Fourier Transform (FFT)
- Send data to the cloud
- Debounce a button
- Use continuity check on a Multimeter (Beepy mode!)
- Convert a signal from analog to digital (ADC)
- Use I2C communication
- Blink an LED
- Commit to a version control system (such as Github)
- Use REPL in MicroPython to blink an LED
- Use UART serial communication
- Use SPI to send/recieve data
- Solder in your own header pins
- Get an Arduino
- Use a Hardware Simulator eg. WokWi

1 tile = 1 point

**Total Score**

START | HERE

Name: _____