SpaceCoastSec Oct 2024

By: Alex Thines

S:i

Structure of the talk

01. Introduce myself
02. Why should you care about this talk?
03. Small (easily acquirable) hacking devices on the market
04. Pros and Cons of each along with prices
05. Go over the different boards that I am a fan of
06. What about other boards???
07. Go over languages and their pros and cons
08. Popular boards used for projects
09. Why to use one of the boards over the others
10. Parts to consider
11. How to get the parts
12. Things to consider when selecting and ordering parts
13. Putting it all together (Sample Project with Workflow)
14. Sick Ideas! Why should I remember this?

S:01

Introduce myself

Senior Penetration Tester specializing in anything that touches web

Likes:

- Messing around with different technologies
- Studying for certifications (I have WAY too many…)
- Programming
- Automating things
- Tinkering with anything I can get my hands on
- Sleeping

Show all 18 licenses & certifications →

Don’t like:

- Being awake before noon
- Net Pen / Social Engineering
- Doing things manually more than 3 times a day

S:02

Why should you care about this talk

- Making your own stuff is fun!
- Last year’s Drone talk used purpose built devices
  - Bringing Watch Dog 2 to life (Using Drones and Arm devices to augment red team engagements) / Guardians of Cybersecurity: Deploying IoT devices via Drones and Dropboxes
- Tinkering with components is a lot of fun!
- Like with programming, it can be useful to make something specific to what you want
- IoT is life
- Cheap vs expensive equipment

S:03

Small hacking devices

Flipper Zero

Price: ~$160

Where to buy:
https://shop.flipperzero.one/

Things to consider:

- Import process
- “Well” known device
- Has a lot of hardware in it
- Public support for specific functionality
- Can expanded via gpio pins

Wifi/USB Nugget

Price: ~$75 (Wifi) / ~$75 (USB)

Where to buy:
https://retia.io/collections/just-nuggets

Things to consider:

- Not as known
- Looks like a toy (More than flipper)
- Does not have as much hardware
- Can expanded via gpio pins

S:04

Pros and Cons of each along with price points of each device

ESP32/8266

- Price:
  - Amazon - $16/3
  - Micro Center - $25
- Power Needed: 5V
- Wireless Functionality:
  - Wifi
  - Bluetooth
- Place buy from:
  - Amazon

Raspberry Pico

- Price:
  - Amazon
    - $22/4 (Not W)
    - $18/2 (W)
  - Micro Center
    - $4 (Not W)
    - $6 (W)
- Power Needed: 3.3V
- Wireless Functionality:
  - Wifi/Bluetooth on W
- Place buy from:
  - Micro Center

Arduino Nano

- Price: $17/3
- Power Needed: 5V
- Wireless Functionality:
  - None (Need to get a board with wireless features added or Nano ESP32)
- Place buy from:
  - Amazon

S:05

What board should I get???

ESP32/8266 vs Raspberry Pico vs Arduino Nano

Personally,

- ESP for complex projects
- Pico for simple projects

Arduino Nano has never interested me

I don’t see them around a lot (Can order them online however)

S:06

What about Pi, Pi Zero, Zima, Uno, etc

Why I don’t use them:

- Size
- Power
- Microcontroller vs Microcomputer

When I would use them:

- Complex attacks
- Don’t want to make tools
- Don’t want to solder something
- USB device is better (Antenna for Wifi attacks, simple GPS module, etc)

What language should I use ???

Python based

MicroPython / CircuitPython

Pros:

- Easier to code since python is common
- Can update code without re-compiling a binary
- Boot.py and Main.py are simple to keep track of
- Easy to interact with internal filesystem

Cons:

- Slower
- Not as much documentation for certain components
- No compile protections / Easier to brick a device
- Coding is harder to do with VSCode than normal

Arduino

Pros:

- Faster
- Compiled protections
- Much more documentation for components
- Very close to C so easy transitioning to C
- IDE experience is a LOT nicer

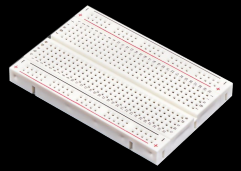
Cons:

- C is more “difficult” to code in vs python
- Updating code on the fly is harder

S:08

Popular boards used for projects

Bread Board



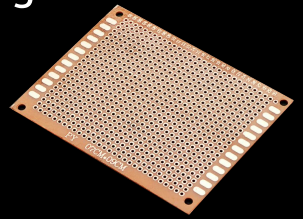
Pros

- Easiest to use
- Can be linked together to make larger (Useful for ESP32s)
- No soldering required
- Cheap (6 piece set for $9 on amazon)
- Lower “skill ceiling”

Cons

- Components not secured
- Bulky

Perf Board



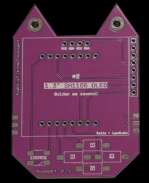
Pros

- Secures parts to board
- Can be made smaller than a breadboard
- Still cheap and comes in many colors
- Looks “more” professional and less prototype-like

Cons

- Soldering required
- Some boards do NOT have paths
- Higher “skill ceiling”

Fabricated Board



Pros

- Looks professional
- Almost completely customizable (Depends on who you use)
- Parts have a place to go and be secured
- Not as low but still lower “skill ceiling”

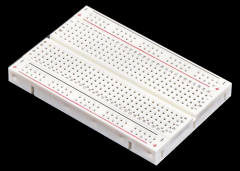
Cons

- Soldering still required
- More expensive
- Requires planned out schematic
- Can’t be made and used “instantly”



S:09

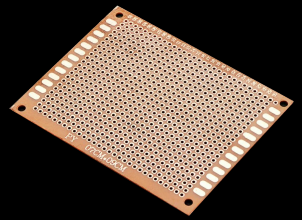
Why to use one of the boards over the others



Early stages of development

Quick Proof of Concept

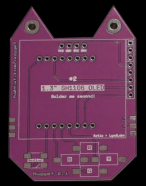
Temporary idea/project and want to reuse components



Require the components to be more secure

Want something that can be deployed quickly

Looks are less important than rapid functionality



Want components to be secure and in a specific spot

Want a more polished look

Specific design/shape/details wanted

You have the board and device! Now what??

Questions to ask:

- How will I get data from the device?
  - Display Module
  - Wifi
  - LoRa
  - LEDs
- How will I interact with the device?
  - Buttons
  - Touch screen
  - Joystick
- Am I trying to measure something?
  - Distance
  - Temperature / Humidity / Light level (Lumens) / Presence
- Am I trying to interact with the environment?
  - Motors
  - IR Blaster
  - Wifi / Bluetooth connections
  - Relays

How to get the parts

Best 2 options for beginners (in my opinion)

1. Order specific parts that you want to use/try out
    a. Amazon
    b. Sparkfun
    c. Adafruit

2. Buy a STEM/Project starter kit
    a. Elegoo
    b. Microcenter / inland

Things to consider when selecting parts

1. DOCUMENTATION
    a. What documentation exists?
    b. Can you find example code for it in your language
    c. Can you figure out how a library works to convert it to your language if the library doesn't exist in your language
2. Is the product end of life
    a. Doesn't always matter but can make finding example code in your language harder
3. Price vs time until arrival
    a. Can you wait that long
        i. Do you want to?
4. Powering the component
    a. Does it match your planned power input
    b. Does programming it and running it away from the computer match in voltage?

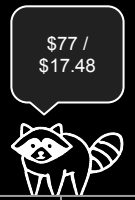
Putting it all together (Sample Project with Workflow)

Decide what you want to make!

- Wants
  - Something to check if I have wireless internet connectivity
  - An indicator if I lose internet
  - Another indicator if I regain internet
- Likes
  - Portable
  - A very easy way to spot if I lose or regain internet
  - Logs (Don’t need them to last)
  - Timestamps

S:13.1

Design Stage



Part	Reason	Price (Amazon)	Price per unit
ESP32	I like ESP more than Pico	$15/3	$5
Micropython	Easiest to do for this imo	Free	Free
Bread Board	Do not need it permanently, it’s ok if it breaks	$7/6	$1.17
SSD1306 screen	Small, simple, like the look	$15/5	$5
Jumper Wires (Male to Male)	Don’t need anything super fancy	$13/100 / 6	$0.78
Small 3.7v Lipo Battery	Makes it portable and 3.7v is enough	$22/4	$5.50
Male header pins (Optional)	Makes a solid connection with board	$5/400 / 2	$0.03

Breaking down the idea

Wants

1. Connect to wireless internet
2. Check if I have connectivity
3. See Status
4. Repeat steps 2 and 3 constantly and allow for issues

Likes

1. Make it easy to transport (External Power/Screen to see)
2. LEDs???
3. Try to get current time
4. Print the information to serial port
    a. Portability can suffer here

- Wants
    - Something to check if I have wireless internet connectivity
    - An indicator if I lose internet
    - Another indicator if I regain internet
- Likes
    - Portable
    - A very easy way to spot if I lose or regain internet
    - Logs (Don’t need them to last)
    - Timestamps

S:13.3

Proving it’s possible

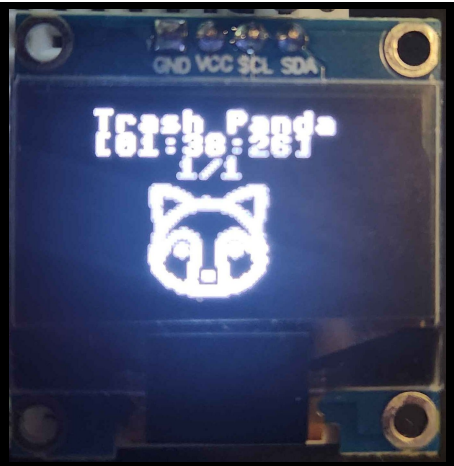
Wants

- Connect to Wifi
- Ping something I know should always be up
  - Get ping (https://gist.github.com/shawwwn/91cc8979e33e82af6d99ec34c38195fb)
- Throw it into a sick while True loop

Likes

- “Paint” the screen
  - Update the screen
- Error handling
  - What if I can’t reach the target?
  - How is the wireless connection handled on break???

```
[+] Operational
[+] Starting wifi
[+] In wifi function
[+] Starting connection process
[+] Attempt 1
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
[+] Connected!
Connected to WiFi: ('192.168.1.241', '255.255.255.0', '192.168.1.1', '172.24.1.69')
PING 8.8.8.8 (8.8.8.8): 64 data bytes
84 bytes from 8.8.8.8: icmp_seq=1, ttl=115, time=154.124999 ms
1 packets transmitted, 1 packets received
[+] Transmitted: 1
[+] Recieved: 1
[01:36:47] 1/1 packets transmitted/received.
```

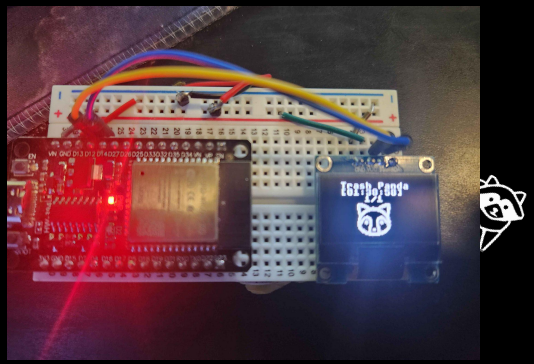


S:13.4

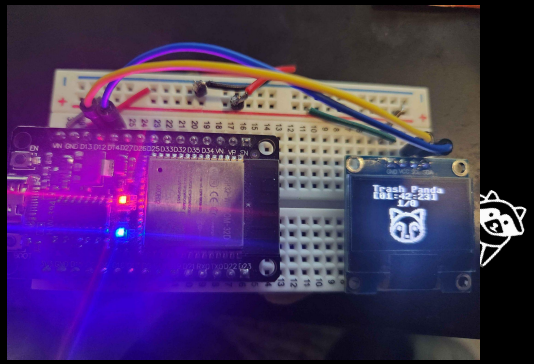
Prototypes

Hardware

Can connect



Can’t connect



Software

Code located at on github

User: sparky23172

Project: Presentations

Subfolders: HackerTool_Talk / pingCheck / main.py

Lines: 211

https://github.com/sparky23172/Presentations/blob/main/HackerTool_Talk/pingCheck/main.py

S:13.5

Upgrades people!!!!

How can this be improved???

- Using a “better” board
- 3D print a case
- Configure syslog to send (UDP 514 connection)
- Actual LEDs
- Configurable by board / Wifi
- Upgrade battery with TP4056 charging module
- Make it a module for a more complicated device (Flipper??)

Sick ideas! Why should I remember this?

Reasons:

1. The walk of 500 miles starts with the first step
2. “Inspections”
3. Special requirements/specifications
    a. Size
    b. Purpose
4. Cool project(s)
    a. OOB communications
    b. War droning/driving/rcing (remote control car)
    c. Spoofer

Thank you!

Questions?