The IoT hacker's swiss army knife

Arduinos

Bruce Barnett
@grymoire

/usr/bin/whoami

- Graduated from RPI in 1973
- 45 years
  - Sr Software Programmer
    - Schlumberger
  - Research Scientist
    - GE, Lockheed Martin
  - Security consultant
    - NYSTEC
- Magician
- Never stop learning

Why this talk?

- This is an intro to embedded electronic development
- IoT devices are proliferating
- IoT devices typically have weak security
- IoT hacking can be done on the cheap
- First step - become familiar with the Arduino
- Fun, not FUD

Thanks to Mark C. AKA @LargeCardinal for the inspiration

What the heck is an Arduino?

- Based on a 2003 thesis project: *Wiring*
- *Problem:*

“… Current prototyping tools for electronics and programming are mostly targeted to engineering, robotics and technical audiences. They are hard to learn, and the programming languages are far from useful in contexts outside a specific technology …”

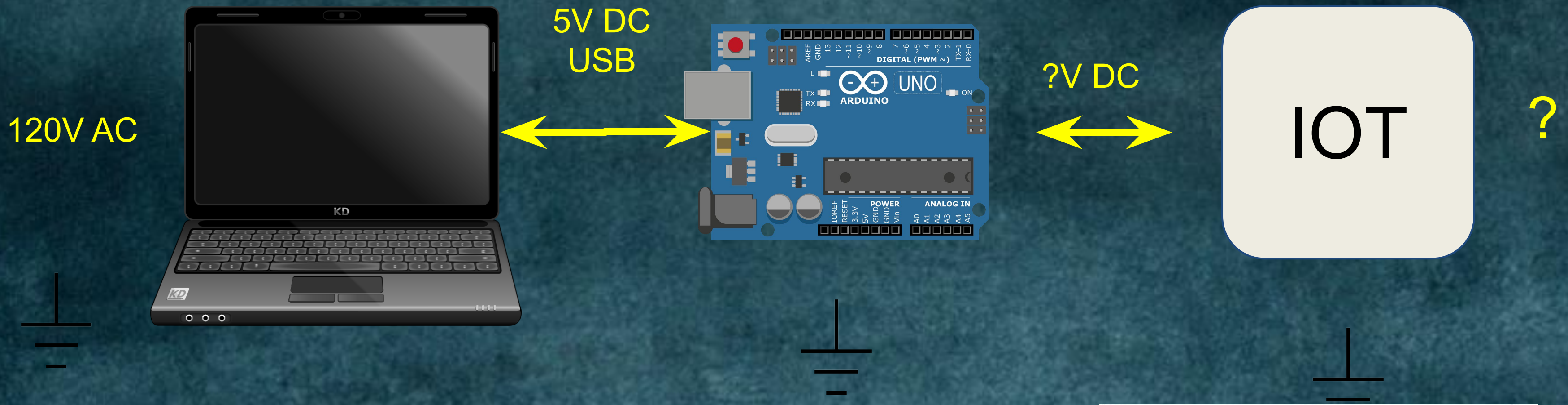
Goal of the Wiring/Arduino project

- “The objective [] was to make it easy for **artists and designers** to work with electronics, by abstracting away the often complicated details of electronics so they can focus on their own objectives.” - Hernando Barragán

Arduino Objectives

- Open Source hardware - i.e. legal clones
- Low cost ($7-$100)
- Flexible
- Easy to use
- Easy to be creative

Connecting data and power



120V AC

5V DC
USB

?V DC

IOT

?

More Power, Igor!

Some precautions

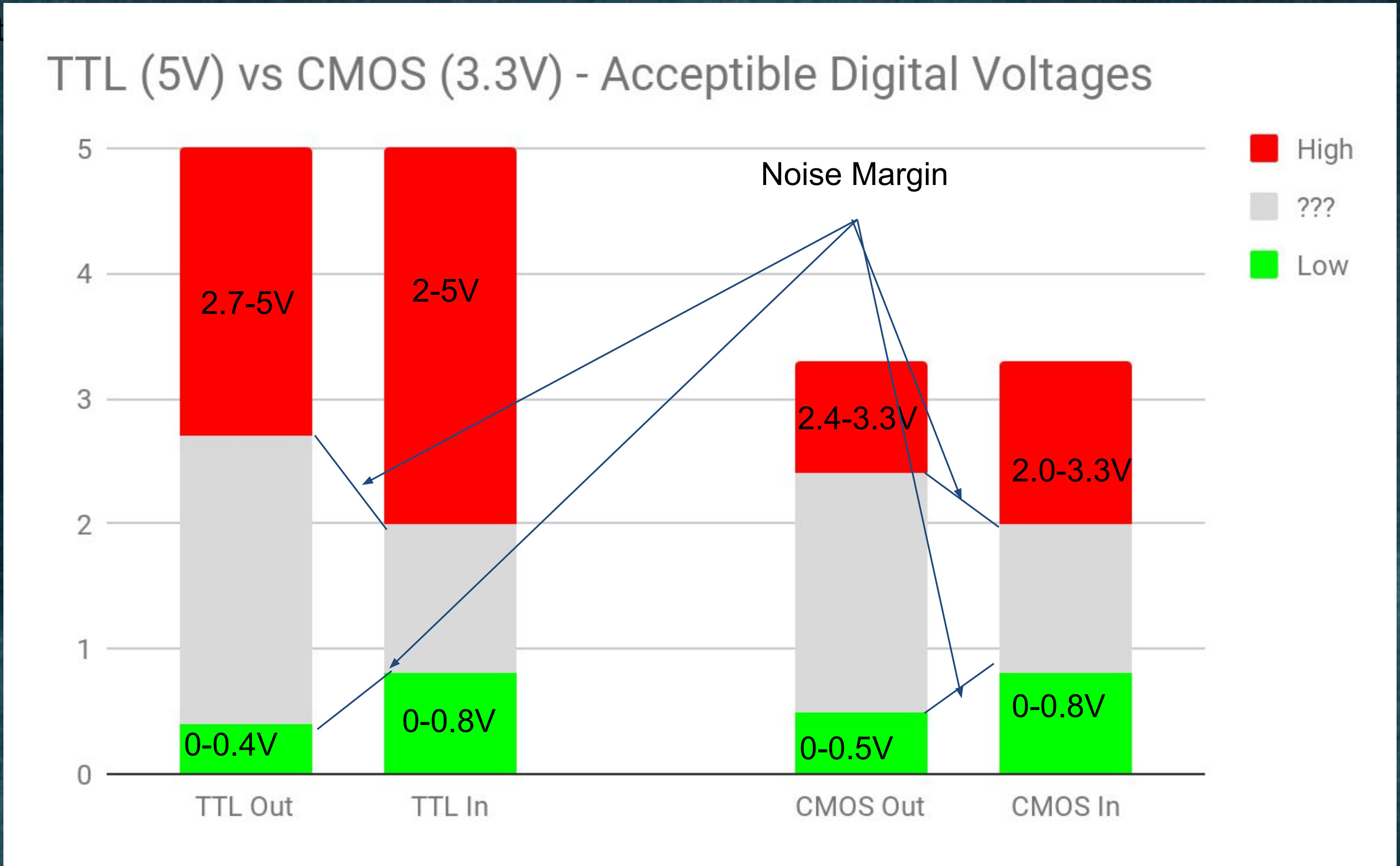
Logic Hardware types

- TTL - Transistor-Transistor Logic
  - 5 Volt
- CMOS - Complementary metal–oxide–semiconductor
  - 3.3 Volt
  - Lower current => batteries last longer
  - Less Heat
  - Smaller, faster

5V vs 3.3V - why is this important?

Digital Voltage Levels

ht



Lesson

Input voltages between valid ranges are indeterminate

- - ¯_(ツ)_/¯

There exists 3.3V and 5V devices and Arduinos

- 5V devices are more robust
- 3.3V is a safe assumption
  - It may not work, but it’s safe
- 5V => 3.3V Fails - maybe

permanently

Grounding inputs

- Connecting to ground is a cheap way to generate a digital “low”
- What happens when it’s not connected to ground?
  - It may “float”
  - Therefore input lines often have a 5K-10K resistor connected to Vcc
    - AKA … a pull-up resistor
- Arduinos often have a programmable pull-up

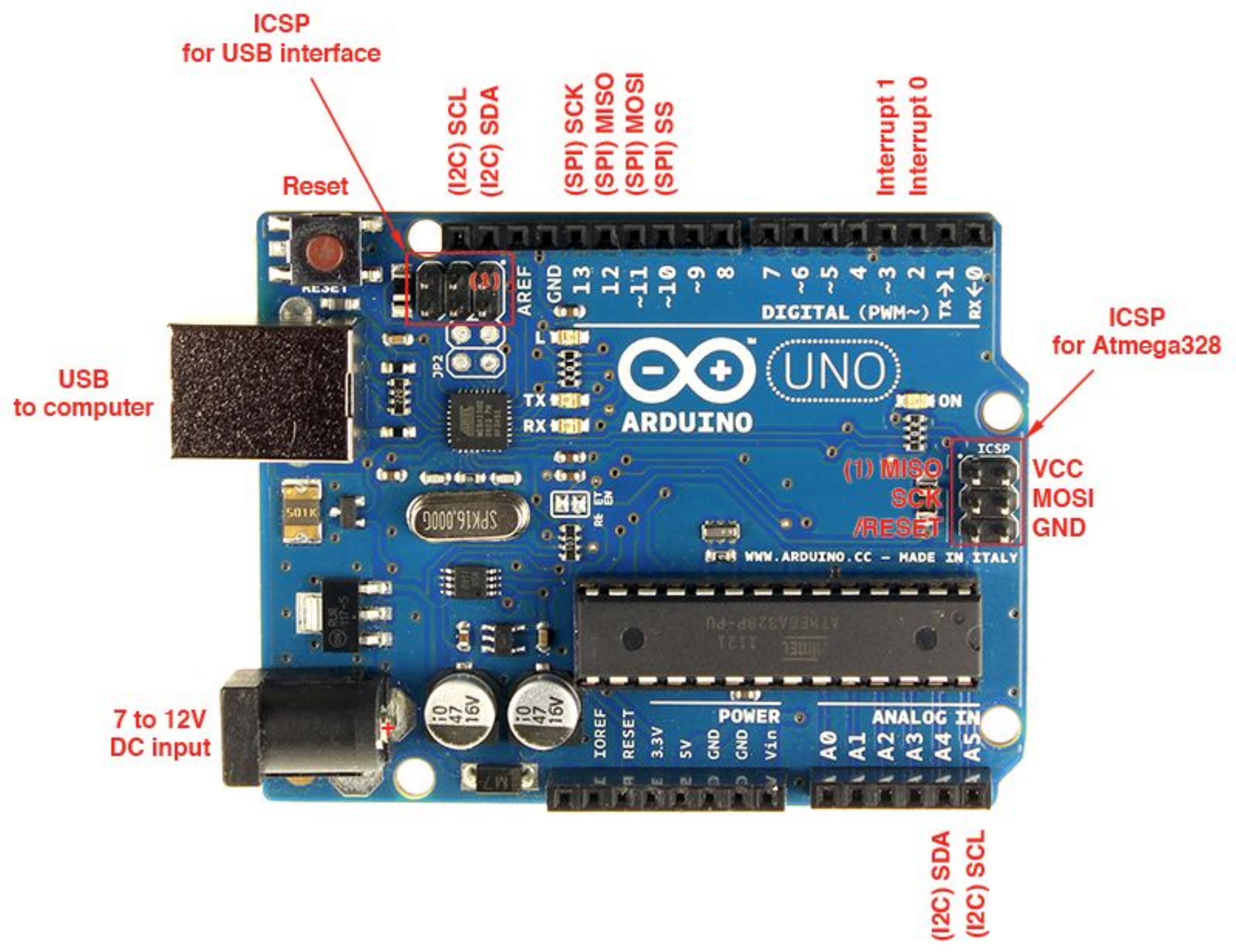
Inputs w/o pullups => ¯_(ツ)_/¯

Grounding Truths

- Separate devices have separate grounds
- 2-prong AC adaptor, powering an Arduino, does not have a ground
- Reverse + and - when connecting can fry electronics
  - USB cable conducts ground into your laptop

Connect ground points first &
Use a Multimeter

Arduino Uno


ICSP
for USB interface
Reset
(I2C) SCL
(I2C) SDA
(SPI) SCK
(SPI) MISO
(SPI) MOSI
(SPI) SS
Interrupt 1
Interrupt 0
USB
to computer
ICSP
for Atmega328
(1) MISO
SCK
/RESET
VCC
MOSI
GND
7 to 12V
DC input
(I2C) SDA
(I2C) SCL
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
UNO
ARDUINO
ON
TX
RX
L
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
IOREF
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5

DIgital I/O Pins

Digital Pins can be set to modes:

```
INPUT              # normally low
INPUT_PULLUP # normally high
OUTPUT
```

Digital Output Values:

- High
- Low

AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
L
TX
RX
ARDUINO
UNO
ON

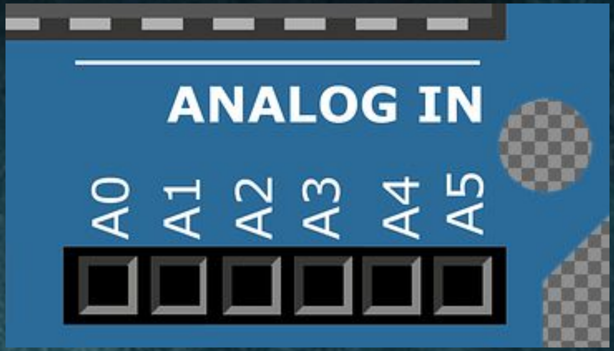
Analog I/O Pins

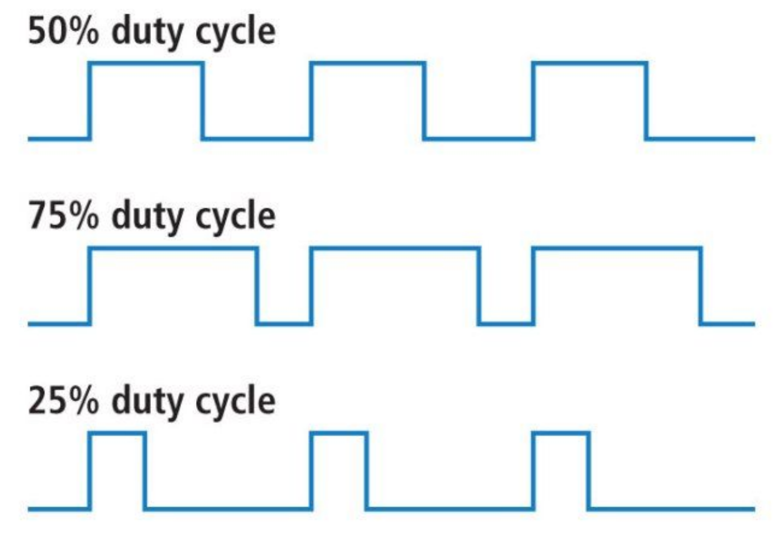
Analog Input:

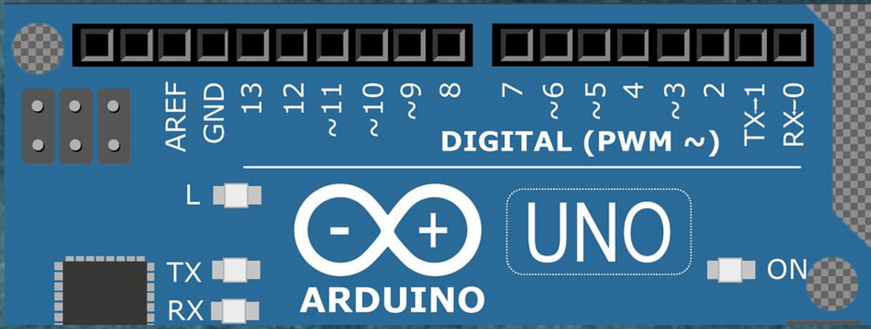
- Analog/Digital convertor (ADC)
- Values from 0 to 255

“Analog” Output:

- Look for pins with ~ after number
- Pulse Width Modulation

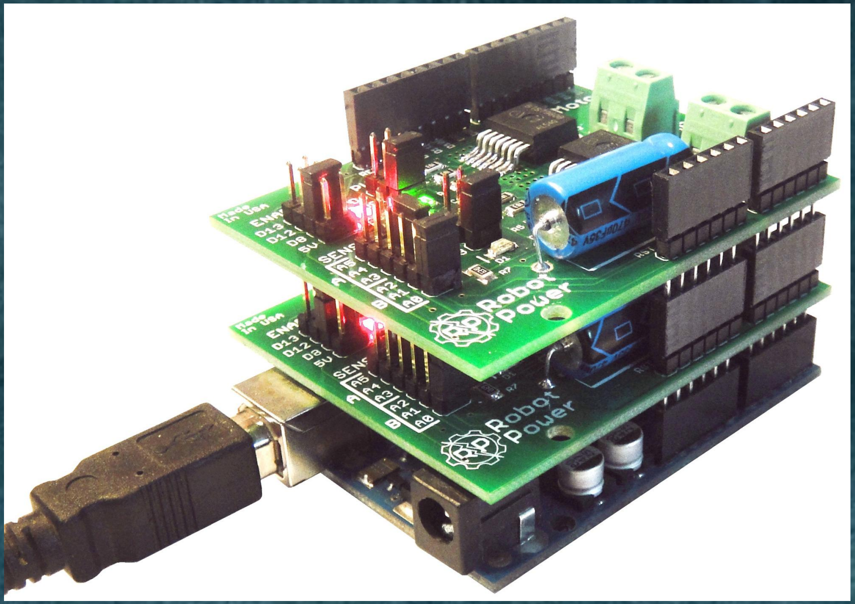




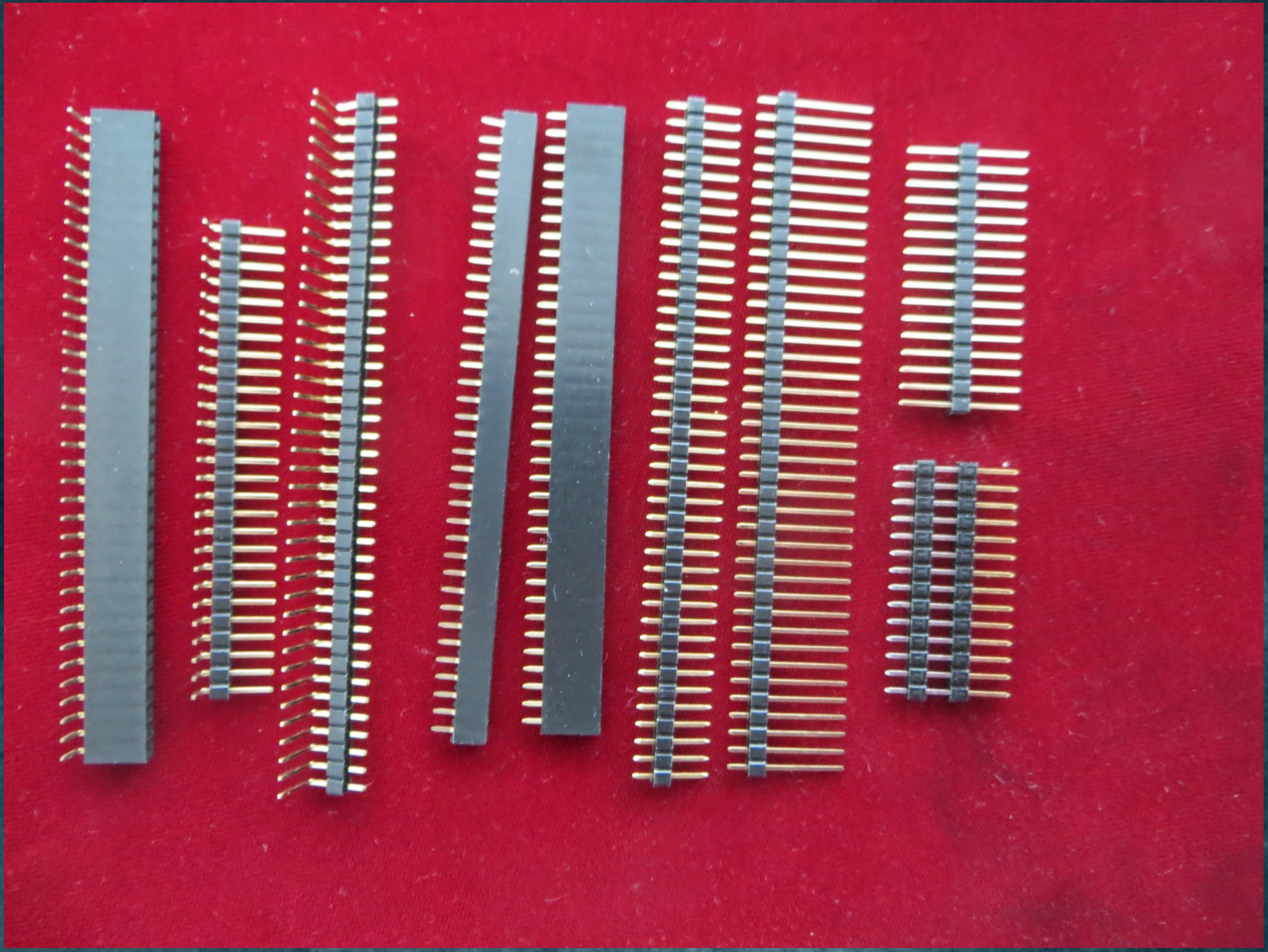


Stackable Arduino Shields

- Ethernet
- Sound
- Motors
- Prototypes
- NFC
- Data Logging
- GPS
- Touch Screens



Headers!



Arduino Software

- Read a digital input (button)

```
int pushButton = 2;

void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  int buttonState = digitalRead(pushButton);
  Serial.println(buttonState); // print to debug console via USB cable
  delay(100);    // delay 1/10th second in between reads for stability
}
```

Installing and Using Arduino IDE

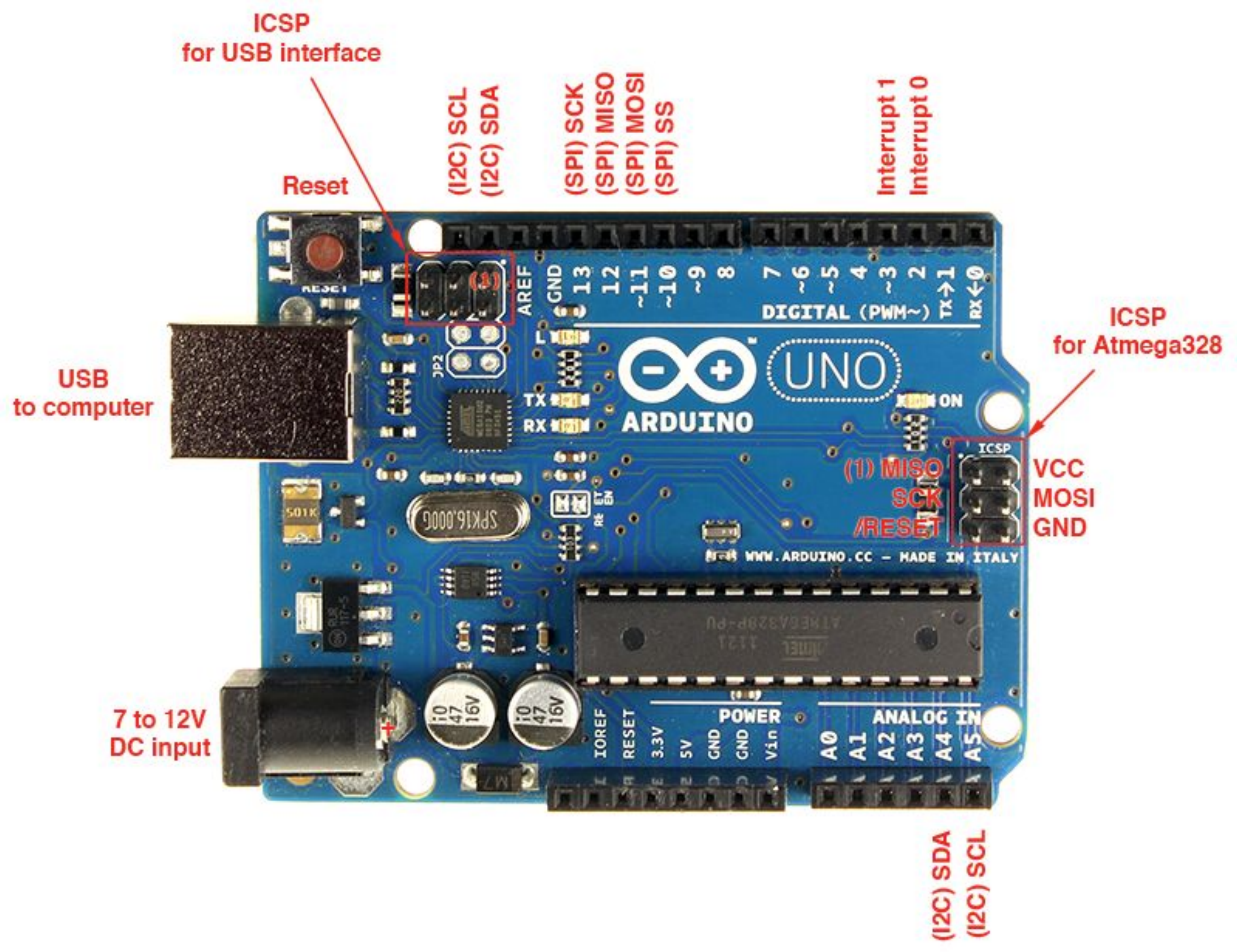
- Download and unpack file
- Execute arduino
- Once running:
    - Select com port
    - Select board type
    - Open “sketch”
    - Edit
    - Verify, upload and run
    - Optionally open debug console

Arduino Demo

Libraries and specialized pins

- Serial/UART
  - Shells, Root consoles
- I^2C - Inter-integrated Circuit
  - Displays, Radios
- SPI - Serial Peripheral Interface
  - Flash Memory, High Speed peripherals, etc
- ICSP - in-circuit serial programming
  - Reprograming Bootloader
  - Arduino-Arduino control
- JTAG - Joint Test Action Group

Arduino Uno


ICSP
for USB interface
Reset
(I2C) SCL
(I2C) SDA
(SPI) SCK
(SPI) MISO
(SPI) MOSI
(SPI) SS
Interrupt 1
Interrupt 0
ICSP
for Atmega328
USB
to computer
(1) MISO
SCK
/RESET
VCC
MOSI
GND
7 to 12V
DC input
(I2C) SDA
(I2C) SCL
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

Serial passthrough

Examples=>04 Communication => SerialPassthrough
Using 2 Hardware Serial ports...

```
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop() {
  if (Serial.available()) {          // If anything comes in Serial (USB),
        Serial1.write(Serial.read());   // read it and send it out Serial1 (pins 0 & 1)
  }

  if (Serial1.available()) {         // If anything comes in Serial1 (pins 0 & 1)
        Serial.write(Serial1.read());   // read it and send it out Serial (USB)
  }
}
```

Serial I/O on any pin

https://www.arduino.cc/en/Tutorial/SoftwareSerialExample

Using Software Serial

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(57600);
  while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.println("Goodnight moon!");
  mySerial.begin(4800);
  mySerial.println("Hello, world?");
}
void loop() { // run over and over
  if (mySerial.available()) {
        Serial.write(mySerial.read());
  }
  if (Serial.available()) {
        mySerial.write(Serial.read());
  }
}
```

Hardware Serial Ports

Arduino	Uno	Mega	Leonardo	DUE	Teensy 3.x
Number of UARTS	1	4	2	4	3

I can either choose an Arduino w/multiple hardware ports, or choose to use SoftwareSerial

HE CHOSE...POORLY

Software Serial

Issues

- Consumes a lot of CPU
- Can’t simultaneous TX/RX
- Lack of options
- Which pins will work for which devices?

Alternates

- AltSoftSerial https://github.com/PaulStoffregen/AltSoftSerial
  - Developed by Teensy Creator
- NeoSWSerial - https://github.com/SlashDevin/NeoSWSerial
  - Only supports 9600, 19200 or 38400 Baud

Get a UART Board

SPI Example - MiFare RFID tag

https://playground.arduino.cc/Learning/MFRC522

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);    // Create MFRC522 instance.

void setup() {
   Serial.begin(9600);    // Initialize serial communications with the PC
   SPI.begin();                    // Init SPI bus
   mfrc522.PCD_Init();    // Init MFRC522 card
   Serial.println("Scan PICC to see UID and type...");
}
void loop() {
   // Look for new cards
   if ( ! mfrc522.PICC_IsNewCardPresent()) {
         return;
   }
   // Select one of the cards
   if ( ! mfrc522.PICC_ReadCardSerial()) {
         return;
   }
   // Dump debug info about the card. PICC_HaltA() is automatically called.
   mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

SPI Example - SPI Flash read

https://github.com/Marzogh/SPIFlash

```
include<SPIFlash.h>
uint32_t strAddr;
#define BAUD_RATE 115200
#define RANDPIN A0
SPIFlash flash;
bool readSerialStr(String &inputStr);
void setup() {
  Serial.begin(BAUD_RATE);
  flash.begin();
  randomSeed(analogRead(RANDPIN));
  strAddr = random(0, flash.getCapacity());
  String inputString = "This is a test String";
  flash.writeStr(strAddr, inputString);
  Serial.print(F("Written string: "));
  Serial.println(inputString);
  Serial.print(F("To address: "));
  Serial.println(strAddr);
  String outputString = "";
  if (flash.readStr(strAddr, outputString)) {
    Serial.print(F("Read string: "));
    Serial.println(outputString);
    Serial.print(F("From address: "));
    Serial.println(strAddr);
  }
  while (!flash.eraseSector(strAddr));
}
```

```
void loop() {

}

//Reads a string from Serial
bool readSerialStr(String &inputStr) {
  if (!Serial)
    Serial.begin(115200);
  while (Serial.available()) {
    inputStr = Serial.readStringUntil('\n');
    Serial.println(inputStr);
    return true;
  }
  return false;
}
```

I2C Scanner

https://playground.arduino.cc/Main/I2cScanner

```
#include <Wire.h>
void setup() {
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}
void loop() {
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ )     {
            Wire.beginTransmission(address);
            error = Wire.endTransmission();
            if (error == 0) {
                      Serial.print("I2C device found at address 0x");
                      if (address<16) Serial.print("0");
                      Serial.print(address,HEX);
                      Serial.println("  !");
                      nDevices++;
            } else if (error==4) {
                      Serial.print("Unknown error at address 0x");
                      if (address<16) Serial.print("0");
                      Serial.println(address,HEX);
            }
  }
  if (nDevices == 0)
            Serial.println("No I2C devices found\n");
  else
            Serial.println("done\n");
  delay(5000);              // wait 5 seconds for next scan
}
```

JTAG

- https://github.com/cyphunk/JTAGenum - Build your own JTAGulator
- https://github.com/mrjimenez/JTAG
  - XSVF File Upload - program CPLDs and FPGA
  - XSVF Assembler/Disassembler
  - TAP Debugger

Powering Arduinos

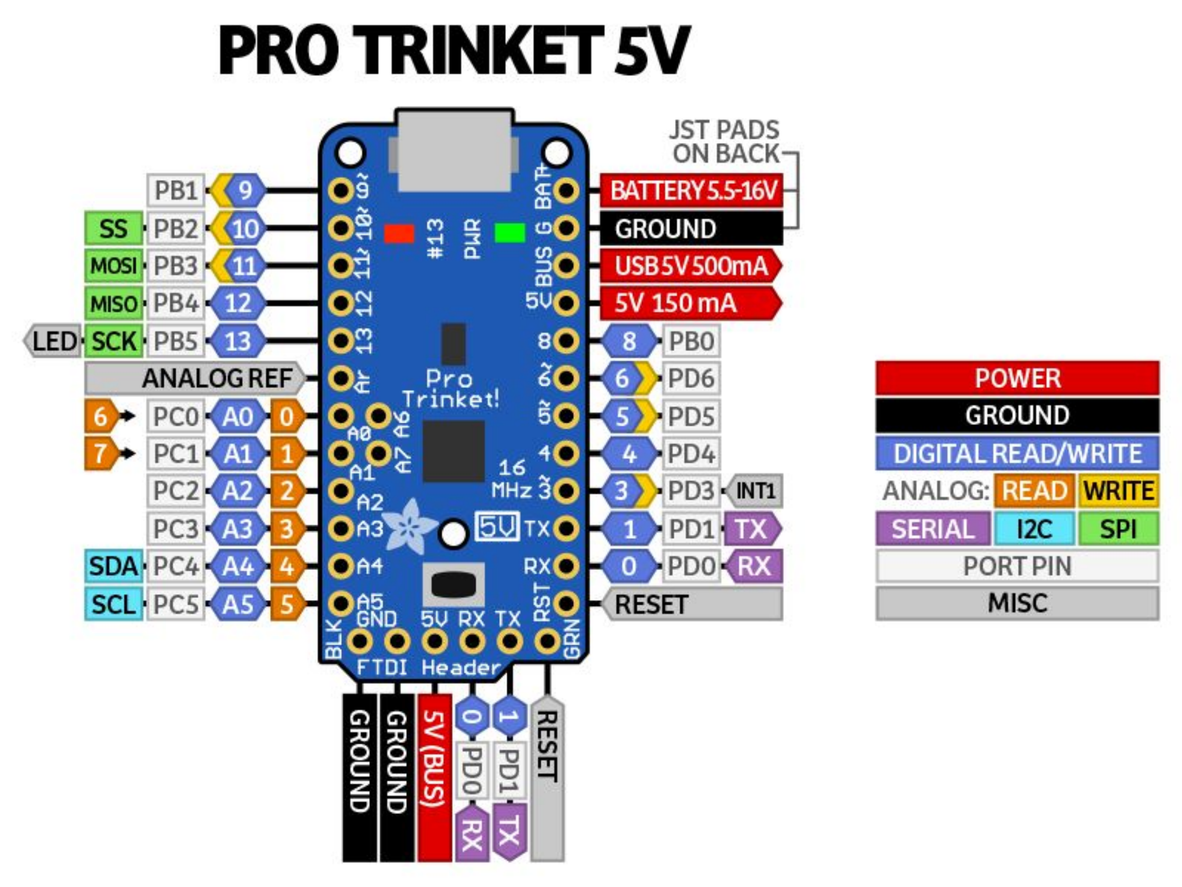
- Barrel Connector (e.g. Uno)
  - AC Adaptor
  - 9V Battery
- USB Cable
  - Computer
  - USB power Pack
- JST connector
  - Battery Pack
  - LiPo Rechargable
- 18650 Battery

Arduino compatible boards

- Favorites
  - Arduino Uno - $25
  - Adafruit Trinket, etc.
  - Teensy
  - ESP32-based
    - D-Duino-32
    - Adafruit HUZZAH32

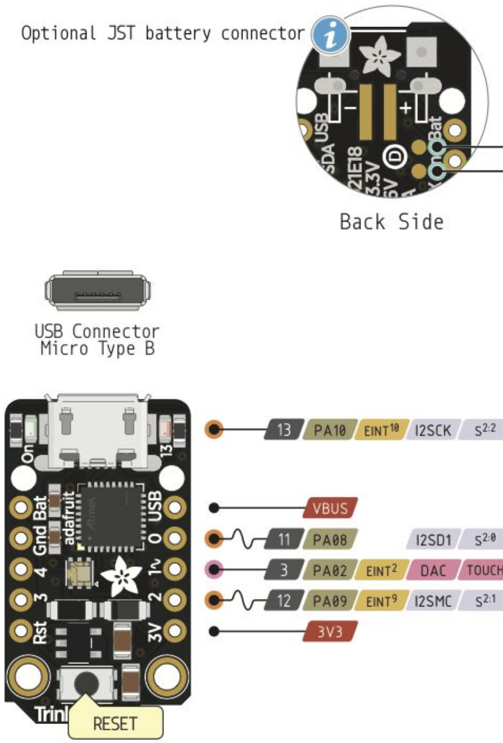
Adafruit Trinket

- $7 - $10
- 5V and 3.3V, Regular, Pro, M0



Trinket M0

PINOUT



DotStar LED

1	PA00	EINT^{0}	S$^{1:0}$	7	DI
2	PA01	EINT^{1}	S$^{1:1}$	8	CI

Flash Access

31	PA30	EINT^{10}	S$^{1:2}$	19	SWCLK
32	PA31	EINT^{11}	S$^{1:3}$	20	SWDIO

🟥	Power	▭	Serial PIN
■	GND	▭	PIN Function
■	Physical PIN	▭	Interrupt PIN
▭	Port PIN	🟧	Control PIN
🟩	Analog PIN	🟪	IDE

∿ PWM Pin

● Port power group

							VBAT		
							GND		
4 A4	AIN6	TOUCH	MOSI	TX	S$^{0:2}$		EINT^{6}	PA06	7 ∿
3 A3	AIN7	TOUCH	SCK	RX	S$^{0:3}$	I2SD0	EINT^{7}	PA07	8 ∿
							RESET	26	

13	PA10	EINT^{10}	I2SCK	S$^{2:2}$	💡			AIN18	13
VBUS									
∿ 11	PA08		I2SD1	S$^{2:0}$	SDA			AIN16	0 A2
3	PA02	EINT^{2}	DAC	TOUCH				AIN0	1 A0
∿ 12	PA09	EINT^{9}	I2SMC	S$^{2:1}$	SCL	MISO		AIN17	2 A1
3V3									

🛇 The total current of each port power group should not exceed 65mA

⚠ Absolute MAX per pin 10mA, 7mA recommended

🛇 Absolute MAX 130mA for the entire package

🛇 GPIO pins rated for 3.3V
Never connect them to 5V signals

VBUS Connected to 5V USB Port
Absolute MAX 500mA

VBAT Positive voltage
from the JST Batt jack

3V3 3V3 output from regulator
Absolute MAX 500mA

adafruit INDUSTRIES

https://www.adafruit.com/product/3500

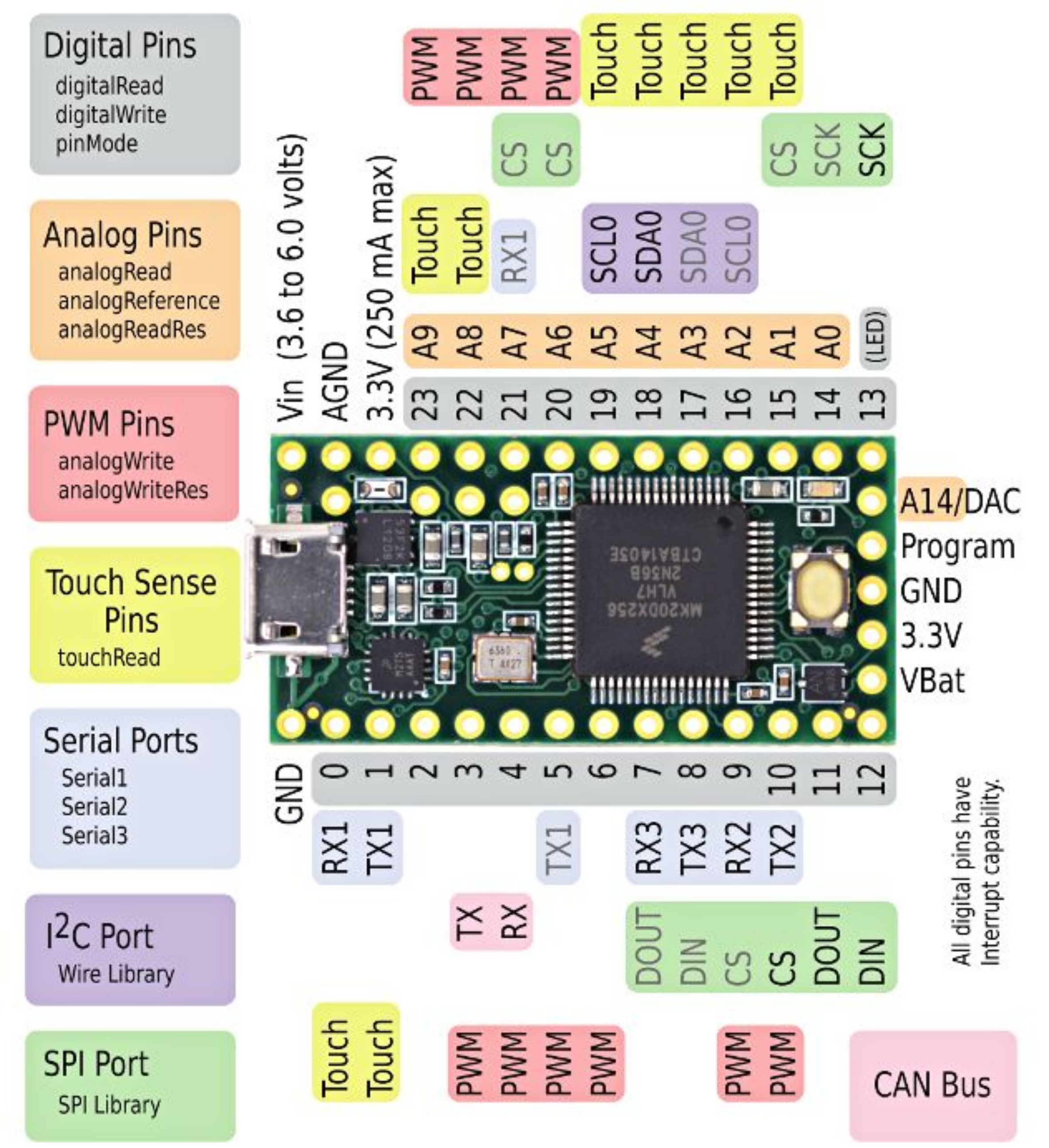
12 SEP 2017
ver 1 rev 0

Teensy

- Small
- HID library
- $12-$35

Teensy 3.2 ($20)

https://www.pjrc.com/



Teensy Projects (or Rubber Ducky)

Google search “teensy pentesting”

- http://www.irongeek.com/i.php?page=security/programmable-hid-usb-keystroke-dongle
- https://matterpreter.com/penteesy/
- https://github.com/samratashok/Kautilya
- https://github.com/Screetsec/Pateensy

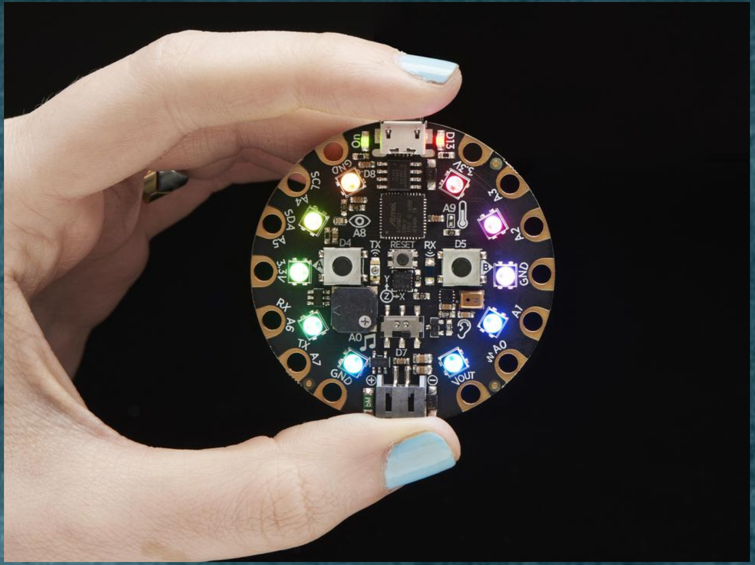
ESP32 boards

- WiFI+Bluetooth included
- Spacehuhn/Travis Lin
  - https://www.tindie.com/stores/lspoplove/
  - $9-$30
  - Preflashed w/WiFi Packet Monitor, Deauther
- Adafruit Feather/HUZZAH32 line
  - $20+
  - 50+ “Wings” - complete line
  - 3*UART, 3*SPI, 2*I2C

And just for fun …..

Adafruit Playground Express - $25

- 10 NeoPixels
- Motion, Temperature, Light, Sound sensors
- Speaker, switch, buttons
- IR transmit/receive (Comm, Prox sensor)
- Touch Sensors, I2C, UART
- IDE's:
  - Arduino
  - CircuitPython
  - MakeCode



Further Adventures

- Other approaches - RPi, BeagleBone
- Special debug hardware
- MicroPython/CircuitPython (i.e. Trinket M0)

```
import board
import digitalio
import time

led = digitalio.DigitalInOut(board.D13)
led.direction = digitalio.Direction.OUTPUT

while True:
        led.value = True
        time.sleep(0.5)
        led.value = False
        time.sleep(0.5)
```

My Favorite Vendors

- Adafruit @adafruit
  - Tuesday they tweet coupon codes
- Teensy/PRJC https://www.pjrc.com
- Seeed Studio - https://www.seeedstudio.com/
- SparkFun - https://www.sparkfun.com/
- Travis Lin (DSTIKE) @dongsentech
  - https://www.tindie.com/stores/lspoplove/
  - w/Stefan Kremser @spacehuhn https://github.com/spacehuhn

Questions?