Using Arduino with TouchDesigner

The NODE Institute SS23

Workshop outline

1. Introduction
2. Arduino Ecosystem, Sensors/Actuators
3. Programming with Arduino
4. Integrating Arduino and TouchDesigner
5. Q&A, etc.

Daniel Molnar (Function Store) - Dan

- Born in **Budapest, Hungary**;
  Living in **Berlin, Germany**
- Masters in Electrical Engineering (2014)
- Software Developer (2014 - 2021)
  - ~5 years as Embedded Engineer
  - ~2.5 years as Audio/DSP Engineer
- …
- TouchDesigner Developer (2022 -)

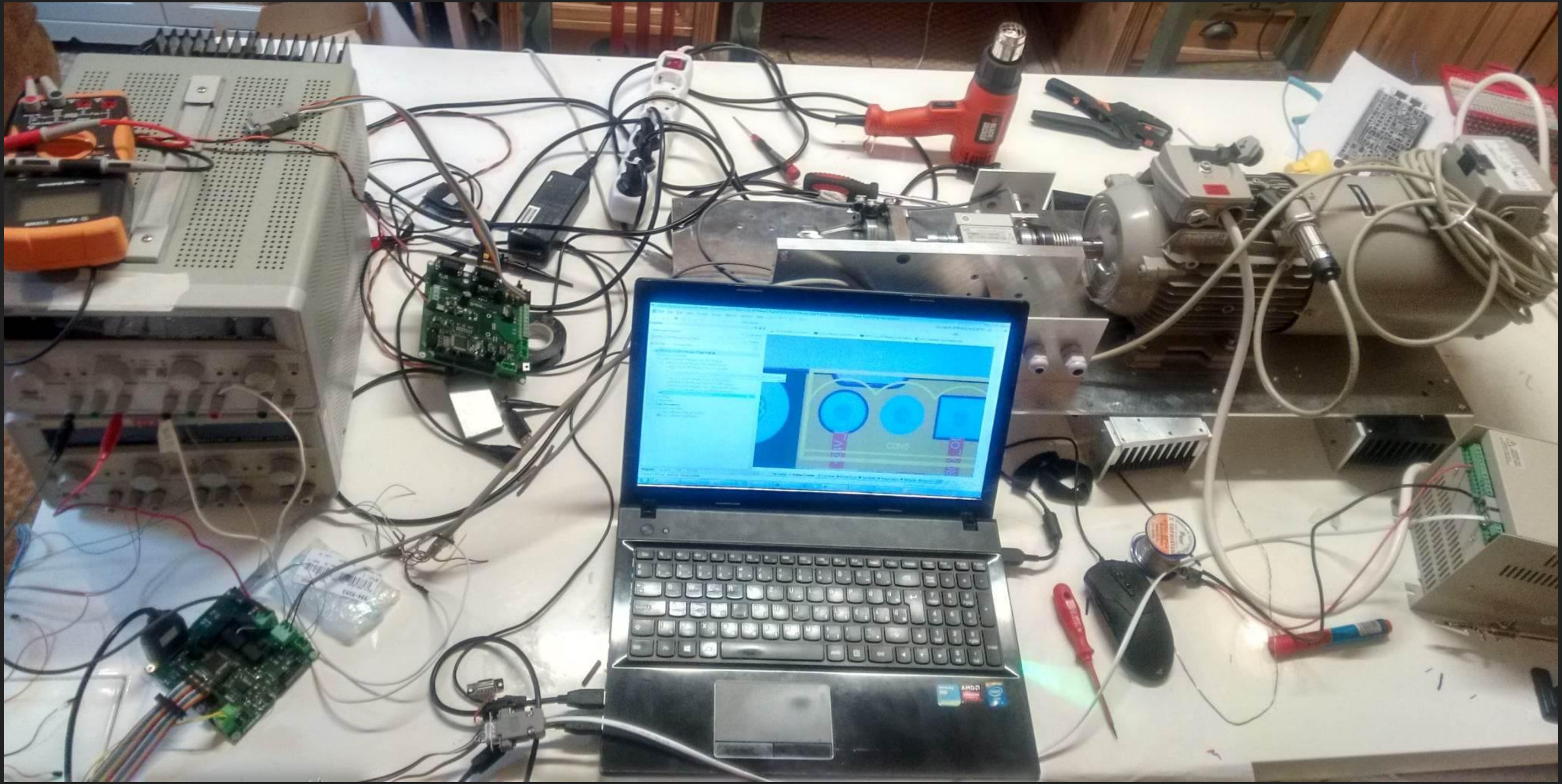
Embedded Life

- Various automotive and consumer electronics projects
  - **Automotive:** brake motor tester system, capacitance tester systems, CAN/modbus controlled relay boxes, test tower monitoring web interface, participation in clutch controller tester development, participation in dashboard display tester development

  - **Consumer electronics:** stereographic DSLR camera control system, professional GoPro operator accessory, iPhone accessory bootloader

  - **Other:** solar panel cleaning robot, smart-metering research and development, pipe leakage monitoring system, participation in railway diagnostics system development

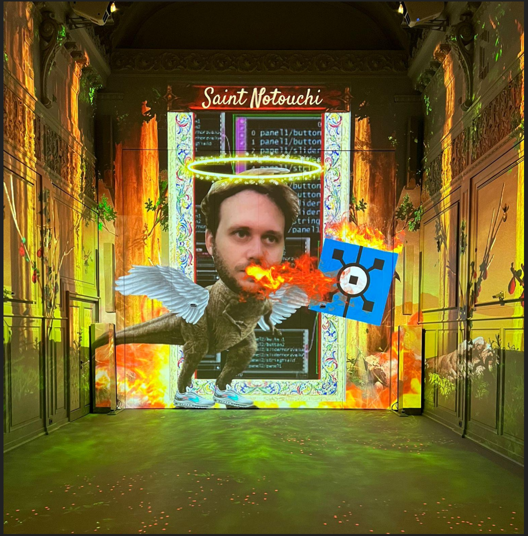
- Custom hardware and software (firmware) stack
- Embedded OS
- Radio communication
- Robust systems



The desk of an Embedded Engineer

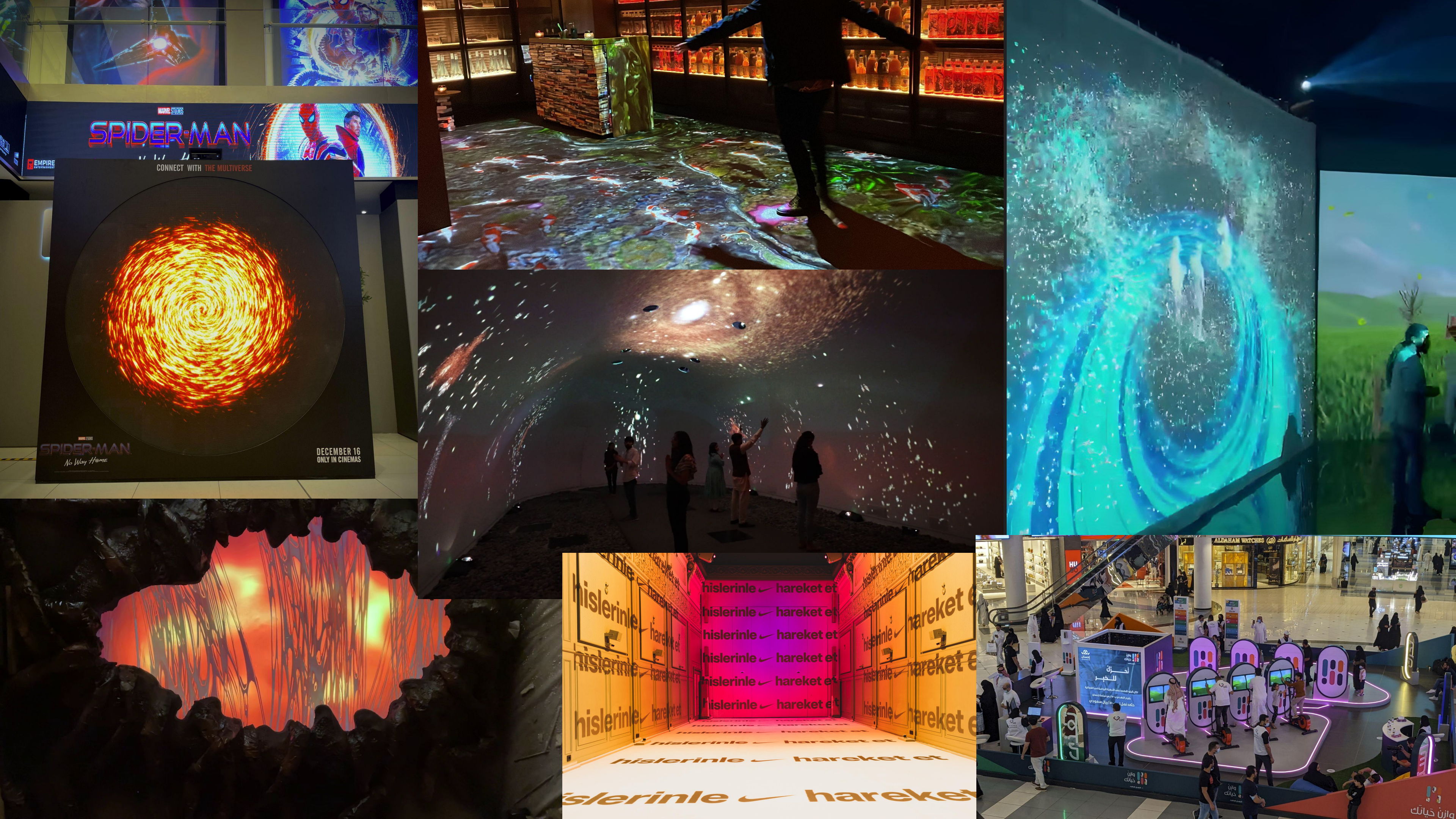
Creative Life

- Music as hobby (guitar/bass)
  - Made visuals in After Effects and Winamp’s Milkdrop2



- Discovered TouchDesigner in 2020
- TouchDesigner Developer (2022 -)
  - Interactive installations, audiovisuals, system design
  - Tool development (YouTube/Patreon)

The desk of a TD developer


MARVEL STUDIOS
SPIDER-MAN
EMPIRE ENTERTAINMENT
CONNECT WITH THE MULTIVERSE
SPIDER-MAN
No Way Home
DECEMBER 16
ONLY IN CINEMAS
hislerinle hareket et
ALDAHAM WATCHES

Inspiration

Kyle Duffield - lots of stuff

Nima Gazestani - Custom Controller

Nima Gazestani - RFID interaction

Irina Petrova - WiFi enabled proximity sensors

Steven Benton - LeapMotion + Servo Motor

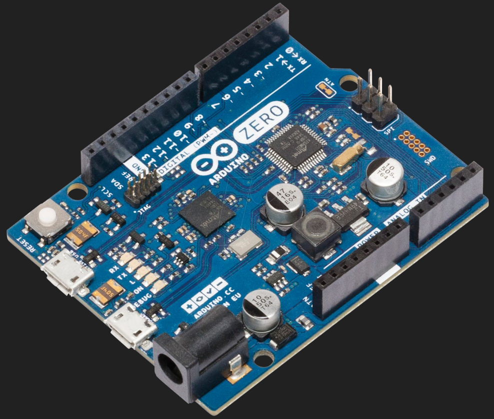
Me (Function Store) - Fitness Activity

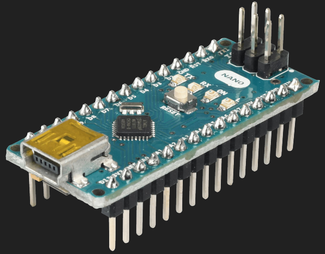
Arduino Ecosystem

- Open-source electronics platform
  - Tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, 2005
  - Hardware and Software for making digital devices and interactive objects
- Aimed at hobbyists
  - Goal: lowering the barrier of entry to electronics and programming
- Professional prototyping
  - Quick and inexpensive way to test an idea
- Great community
  - Countless forums, blogs, tutorials
  - Open-source

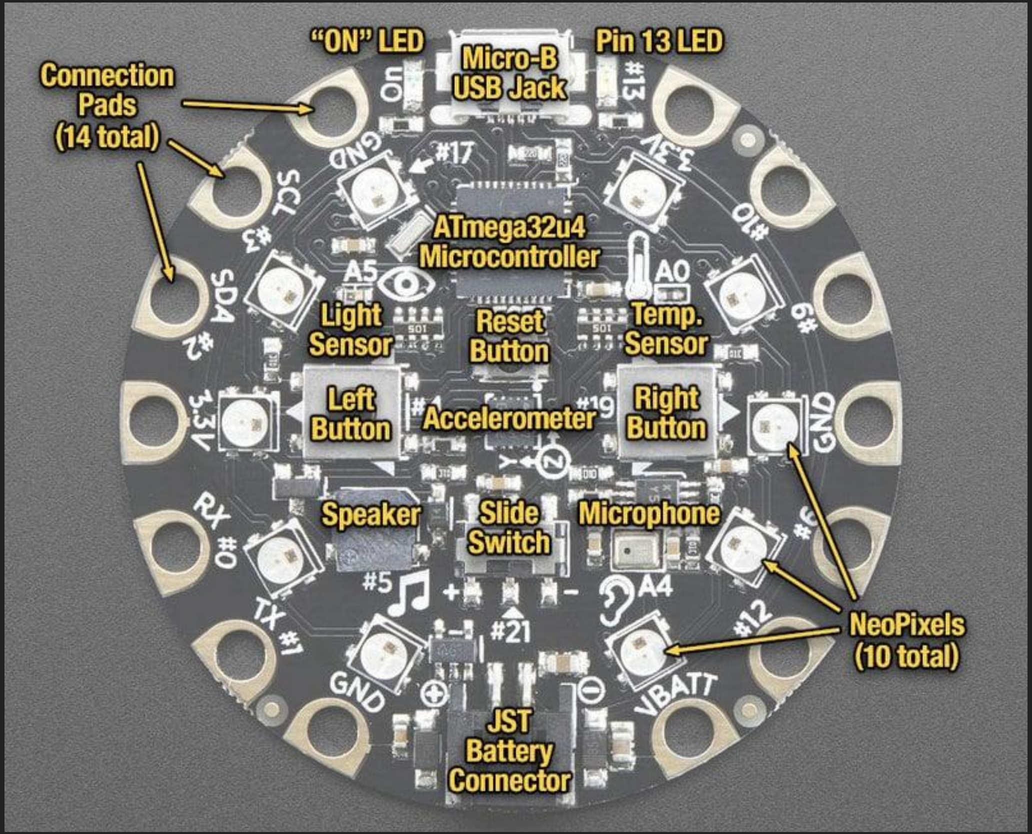
Hardware

- Lots of different boards
  - Usually based around an Atmel microcontroller, varying amounts of flash memory, pins, and features
  - “Arduino-compatible” boards
    - Such as Adafruit Circuit Playground
- Shields (addon boards)
  - Extra functionality modules
  - Stackable
- Plethora of sensors/actuators



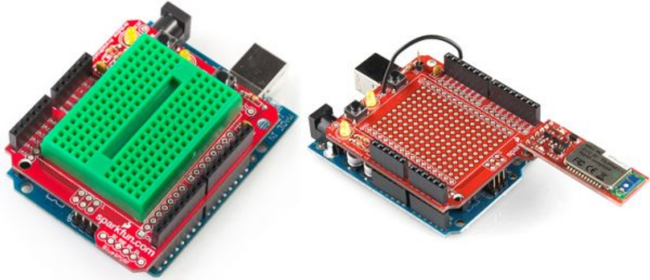


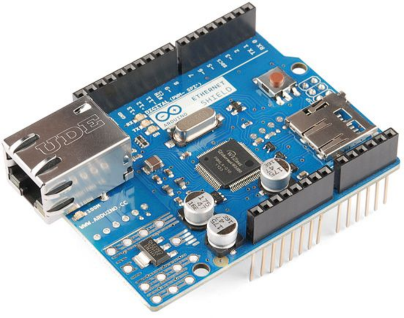
Adafruit Circuit Playground (classic)

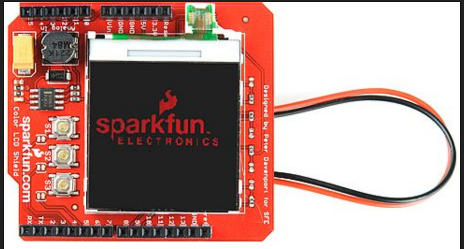


Shields

- Breadboard
- Ethernet
- Motor
- WiFi
- SD Card
- GPS
- GSM
- LCD Screen
- …







Sensors

Basically anything that you can measure physically

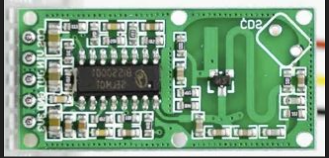
- Temperature*
- Humidity
- Ultrasonic distance
- Infrared motion
- Light sensor*
- Accelerometer*
- Gyroscope
- Soil moisture
- Capacitive touch*
- Microphone*
- Magnetic Hall
- Buttons
- Potentiometers
- etc

Often you can find “Arduino sensors” designed with the hobbyist market in mind:

- Ease of use with Arduino boards
- Standardized connector
- Conditioning resistors/capacitors/amplifiers
- Sample code / Library
- Eg.: Grove system by Seeed Studio

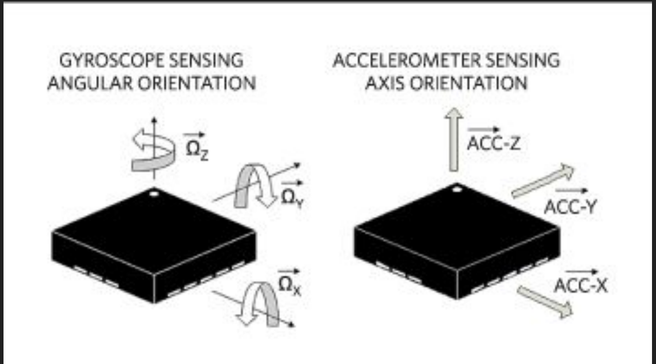
Motion sensors

- PIR Motion Sensor
  - Heat energy from body -> IR radiation
  - Watch out for other IR sources (sun)
  - ~10 bucks, 6 meters range 110°x70° (depends)
  - Good for triggering events in an installation space
  - Outputs low/high
- Microwave Radar Motion Sensor
  - Doppler effect
  - Does not rely on heat
  - Generally bigger range, can go through walls
- Ultrasonic Distance Sensor
  - Time-of-flight principle (time between emitting and receiving an ultrasonic wave)
  - Sound reflection
  - 4m range
  - Usage example



Accelerometer, Gyroscope

- Gyroscope measures the rotation rate around a specific axis (roll, pitch, yaw), helps maintain orientation
  - Steadycam
- Accelerometer measures the rate of change of velocity along XYZ axis. Gravity is always present!
  - Integrating acceleration data over time, velocity can be determined
    - Drift over time!
  - Can derive roll and pitch (see later)
  - Gesture sensing, indoor navigation



Actuators

Interacting with the physical world:

- Servo motors
- DC motors
- Stepper motors
- Relays
- LEDs
- Piezo buzzers
- Solenoids
- Pumps and valves
- …

Similar to sensors you can find “Arduino compatible” actuators designed with the hobbyist market in mind.

Brands to look out for:

- Adafruit
- Sparkfun
- Grove

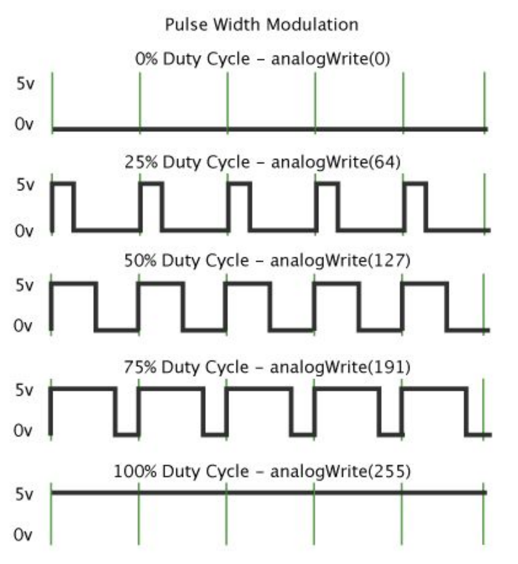
Controlled Motion Devices

AC/DC Motors	Stepper Motors	Servo Motors
Simple direction control	Moves in discrete steps	Moves to position - requires feedback
Less precise than stepper/servo, needs external feedback	Less accurate, can lose steps	Higher precision
High torque, constant at all speeds	High torque at low speed, decreases	~ torque at wide range of speeds, high speed
Cheaper and simpler (without feedback)	Cheaper and easy to control	More complex and expensive
Smooth and quiet	Noisy and generate vibration	More smooth and quiet

Interfaces

How can we communicate with sensors/actuators/other systems?

- **Digital I/O**: reading binary values (HIGH(1) or LOW(0))
- **Analog I/O**: reading voltage of an analog signal (usually 10 bit precision)
- **PWM I/O**: no true analog output but Pulse-Width Modulation can simulate to drive speaker, control brightness of LED or speed of motor


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Serial Interfaces

- UART/”Serial” (Universal Asynchronous Receiver/Transmitter): asynchronous serial communication - computer/GPS/Bluetooth
    - **TX (Transmit)**: This line is used to send data from the UART to another device.
    - **RX (Receive)**: This line is used to receive data sent to the UART from another device.
    - **GND (Ground)**: This line is used as a common ground reference between the two communicating devices.

- SPI (Serial Peripheral Interface): synchronous data transfer to one or more devices over short distances, high speed
    - **SCLK**: Serial Clock (output from master)
    - **MOSI**: Master Out Slave In (data output from master)
    - **MISO**: Master In Slave Out (data output from slave)
    - **SS/CS**: Slave Select (often active low, output from master to indicate that data is being sent)

Serial Interfaces

- **I^{2}C (Inter-Integrated Circuit)**: sync comm of multiple slaves to a master device, simple and efficient, minimal pin requirement
  - **SDA (Serial Data)**: The line for the master and slave to send and receive data.
  - **SCL (Serial Clock)**: The line that carries the clock signal.
- **CAN (Controller Area Network)**: message-based protocol, automotive/industrial equipment

Comparison with Other Microcontrollers

- **Raspberry Pi**: Technically a single-board computer (SBC). More powerful processing capabilities, full operating system for complex software-intensive projects. Less straightforward handling of low-level hardware tasks, more power requirement.
- **ESP Family**: WiFi/BLE capable more powerful microcontrollers, IoT projects. Compatibility with Arduino.
- **Standalone AVR and ARM uC’s**: Programmable with Arduino IDE. Lower cost for mass production, more hardware complexity.
- **Commercial uC Boards**: Lots of commercial products like the STM32 series with higher performance and more features, but less beginner-friendly.

Resources

- https://docs.arduino.cc/learn
- https://docs.arduino.cc/built-in-examples
- https://projecthub.arduino.cc/
- https://www.instructables.com/search/?q=arduino&projects=all
- https://www.tinkercad.com/ - simulator
- https://wokwi.com/ - simulator
- https://github.com/adafruit/Adafruit_CircuitPlayground
- https://td-olib.org/
- https://derivative.ca/community-post/asset/wavetop/67796