SEICHE 2022
Introduction to Arduino
Lesson 2 – Intro Part 2

Instructor: Paul Frommeyer

Cor[orate S[onsors

DXC Technology

PFC Networks

Lesson Plan Overview

Lesson 1 – Intro and Setup
[may require 2 classes]
- Introduction to class format
- Overview of lesson plan
- Presentation format (monitor, camera, screen, whiteboard)
- Inventory of Arduino kits
- Inventory of USB drives
- Installation of ESP32 serial port drivers (Windows only)
- Installation of Arduino software
- History of Computing

Lesson 2 – Laptop operation review
- Control panel/Settings location
- Home directories and folder hierarchy
- Arduino file locations
- Search functions
- Open questions and issues
- History of Telecommunications and Networking

Lesson 3 – IDE essentials
- Setting board type
- Loading example sketches
- Finding and selecting the Arduino USB port
- Uploading a sketch to the microcontroller
- Basic Arduino sketch (program) structure

Lesson 4 – Computer Data Formats
- Binary math
- Basic data types

Lesson 5 – DC Electricity Basics and DMM Intro
- Quick review: DC vs AC electric currents
- DC current operation
- Voltage vs Amperage (current)
- Introduction to basic DMM functions and usage
- Series vs parallel circuits
- Ohm's law

Lesson 6 – Elementary Components
- Resistors
- Diodes and LED's
- Capacitors: Electrolytic (polarized) vs non-electrolytic (non-polarized)
- Simplified transistor operation

Lesson 7 – ESP32 External power sources [may require 2 classes]
- Microcontroller board power requirements
- Board power input options
- Components with different voltage requirements
- External power circuit design

Lesson 8 – Connecting LED's to the ESP32 controller board
- Microcontroller pinouts and diagrams
- Different types of pins
- Analog input
- Digital Input
- Digital output
- LEDs, constant current, and dropping resistors

Lesson 9 – Connecting smart LED strips
- Simplified serial communication protocols
- One-wire serial communication protocol
- Hardware vs software serial
- WS2812B and Neopixel connections
- External device power design

Lesson 10 – Connecting Sensors
- I2C serial protocol
- I2C connections
- I2C sensor board power design
- I2C sensor external power design

Lesson 11 – Connecting LCD and Matrix Displays
- The SPI serial protocol
- SPI connections
- SPI device external power design

Lesson 12 – Connecting other sensors [time permitting]
- Connecting buttons
- Connecting potentiometers
- Reading buttons
- Reading potentiometers

“Lesson” 1 – Intro And Setu[

Part B

- USB Flash Drive Inventory
- Review of Laptop GUI operation
- CP2102 Driver Installation (Windows only)
- Arduino IDE Software Installation
- History of Computing

USB Flash Drive Inventory

DOCUMENTATION – Arduino Reference
- Programming with Arduino.pdf
- Arduino Cookbook-2ndEdition.pdf
- Arduino-For-Beginners-REV2.pdf
- IntroArduinoBook-AlanSmith.pdf
- IntroductionToArduino-Book-AlanGSmith.pdf
- Make_Getting_Started_with_Arduino_3E.pdf

DOCUMENTATION – Electronics Reference
- the-original-guide-to-boards-2021.pdf
- Basic Electronics-Semiconductors.pdf
- Grobs Basic Electronics 2010.pdf
- Instructables-Basic-Electronics.pdf
- Intro to Electronics-Noisemantra.pdf
- Make Electronics 2nd Edition by Charles Platt.pdf
- SPIE-TT107-PracticalElectronicsforOpticalDesignandEngineering-Chapter1.pdf

DOCUMENTATION – WiFi Kit 32 Reference
- Quick review: DC vs AC electric currents
- WIFI_Kit_32_Schematic_diagram_V2.1.PDF
- HeltecWifiKit32-ReferenceURLs.txt
- WIFI_Kit_32_pinoutDiagram_V2.1.pdf

DOCUMENTATION – Boards Guides
- Original Boards Guides from 2019-2022

DOCUMENTATION – Class Information
- SEICHE-IntroToArduino-Lesson1-V1.pdf
- IntroductionToArduino-ElectronicsKitInventory-TextFormatWithDescriptions.txt
- IntroductionToArduino-LessonPlan-8Dec21-V1-PLF.pdf
- SEICHE-ArduinoClass-Summary-V1.txt
- SEICHEArduinoKitInventory-Brief.txt

Critical Documents
- Make Getting Started With Arduino – Use this for self study and back reference
- Arduino Cookbook 2nd Edition – Use this for reference and advanced study
- Make Electronics 2nd Edition – Use this for self study and back reference
- Heltec WiFi Kit 32 Pinout – Maybe print this out since you'll be referring to it often
- Boards guides – If you want to select and purchase a board for your own use
- Electronics Kit Inventory – To keep track of items or if you think you may have lost something

RETURN FLASH DRIVES AT END OF EACH LESSON
- I'll load new content each week, at least that week's lesson but also sketches

La[to[GUI O[eration

- Desktop and task bar
- Start menu and control/settings panels
- Launching applications
- Context sensitive menus (right click)
- Filesystem browsers
- File and folder hierarchies
- Searching for applications
- Accessign and safely removing flash drives

Outside class scope: Application usage other than Arduino IDE, operating system customization, anything command line or related

Windows – CP2102 Driver Installation

- Navigate to software folder on USB drive
- Open Windows folder, then folder for your OS
- Open Release Notes and follow the instructions
- We can walk through some installs, and screen share if desired so everyone can see how this works
- MacOS X probably doesn’t need a driver install, but it’s on the USB if needed
- Kubuntu does not need an install, drivers are in kernel
- Believe it or not, this is probably the trickiest part of the entire class

If you are at all not sure what you’re doing, or what Windows version or architecture you’re using (e.g. x86 vs x64) then wait for assistance

Arduino IDE Installation

- Navigate to software folder on USB drive
- Open folder for your OS
- Windows you can double click the installer
- MacOS X drag Arduino.app to your Applications folder
- Kubuntu
  - Open Konsole
  - Enter “`lsblk`” to find your USB drive
  - Enter “`tar xvf usbdriveidentifier/Software/Linux/arduino1.8.19-linux64.tar`”
  - Enter “`cd arduino1.8.19-linux64`”
  - Enter “`sudo ./install.sh`” and enter your password when prompted

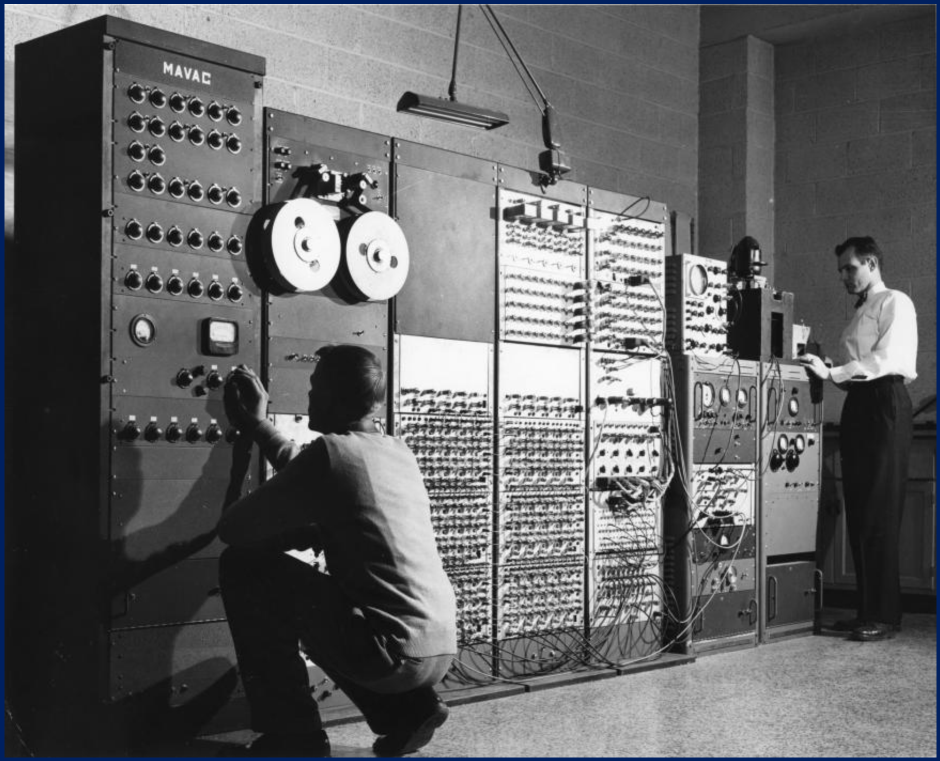
WE WILL DO BOARD SETUP NEXT WEEK

If you are not sure what you’re doing, then wait for assistance or ask for help

(This part is not as tricky as the CP2102 driver installation)

History of Com[uting

- The Jacquard Loom
- The Analytical Engine
- ENIAC and MULTIVAC
- The mainframe era
- The minicomputer era
- The microcomputer era
- The market mitosis
  - Microcomputers
  - Microcontrollers
- Microcomputer evolution
- Microcontroller evolution

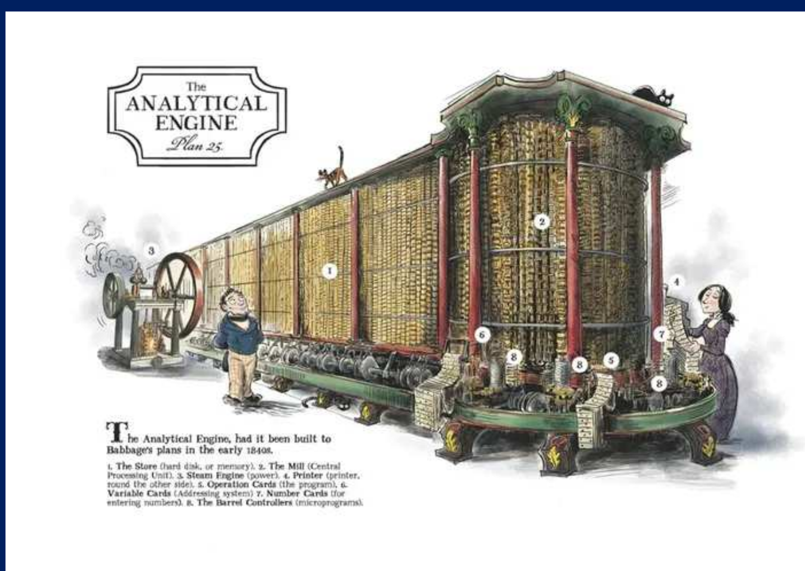


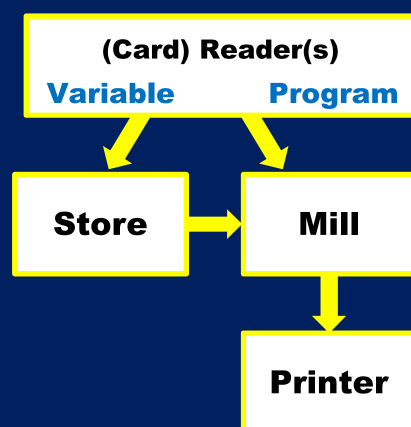
Jacquard Loom

- Invented by Joseph Marie Jacquard in 1804
- Technically, a Jacquard *Machine* which was attached to the heddles of an existing pattern loom
- Provided automatic control of the heddles managing the warp threads
- Used large punched cards to define the woven pattern
- A hole in the card would cause that thread of the warp to be raised, thus excluded from the weft during a shuttle pass
- Allowed extremely complex patterns to be woven
- Once machine driven, gave rise to modern weaving industry
- Mechanism still in use today, but electromechanical relay driven and controlled by digital computer

Babbage Analytical Engine

- Invented by Charles Babbage, 1837
- Construction was never completed
- Four parts: Reader, store, mill, and printer
- While using mechanical parts, was a programmable general purpose computer in the sense we understand
- Considered a landmark invention in computer science

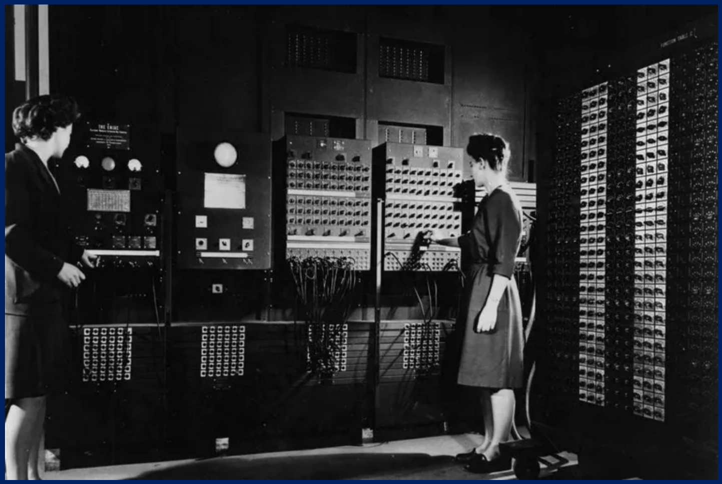




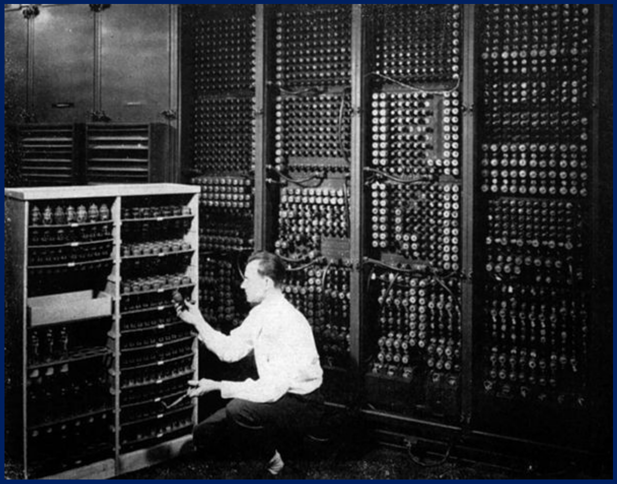
Part of the Engine

Early Com[uters – ENIAC

- ENIAC - Electronic Numerical Integrator and Computer – was the first programmable, electronic, general-purpose digital computer made in 1945
- By the end of its operation in 1956, ENIAC contained 18,000 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors, and approximately 5,000,000 hand-soldered joints. It weighed more than 30 short tons (27 t), was roughly 8 ft × 3 ft × 100 ft (2 m × 1 m × 30 m) in size, occupied 1,800 sq ft (170 m2) and consumed 150 kW of electricity.



ENIAC o[erator's control board



Part of ENIAC

Early Com[uters – UNIVAC

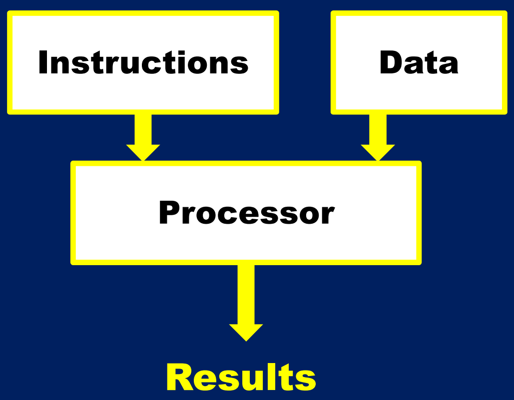
- UNIVAC I (UNIVersal Automatic Computer I) was the first general-purpose electronic digital computer design for business application produced in the United States
- UNIVAC I used about 5,000 vacuum tubes,weighed 16,686 pounds, consumed 125 kW, and could perform about 1,905 operations per second running on a 2.25 MHz clock
- Besides the operator's console, the only I/O devices connected to the UNIVAC I were up to 10 UNISERVO tape drives, a Remington Standard electric typewriter and a Tektronix oscilloscope. The UNISERVO was the first commercial computer tape drive commercially sold
- Evolutionary rather than revolutionary

UNIVAC – Classic diesel[unk com[uting

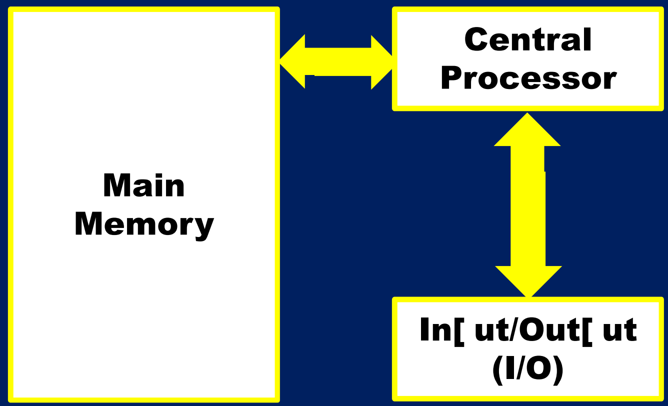
UNIVAC O[erator’s Console

Von Neumann Architecture

- Earliest computers had fixed (hardwired) programs
- Control storage (program) was originally separate from data storage (information to be operated upon)
- John Von Neumann invented the idea of using a single storage medium for both instructions and data
- This was a revolutionary design, reducing costs and increasing flexibility, and is used by all computers since



Historic Architecture



Von Neumann Architecture

The Mainframe Era

- From 1952 into the late 1960s, IBM manufactured and marketed several large computer models, known as the IBM 700/7000 series
- The IBM System/360 (S/360) is a family of mainframe computer systems that was announced by IBM on April 7, 1964, and delivered between 1965 and 1978.[1] It was the first family of computers designed to cover the complete range of applications, from small to large, both commercial and scientific. The design made a clear distinction between architecture and implementation, allowing IBM to release a suite of compatible designs at different prices.
- The mainframe era saw the transition from tubes to transistors, yielding massive economies of scale and increases in computing power
- Also transition of storage from cumbersome mercury based to magnetic core memory

IBM 700 Series

IBM System/360 CPU

S/360 CPU and Peri[herals

The Minicom[uter Era

- Started by Digital Equipment corporation in 1963
- The PDP-5 was Digital Equipment Corporation's first 12-bit computer
- Followed by the PDP-8 and the hugely popular PDP-11
- Culminating in the “mini-Mainframe” VAX-11/780
- Transition from transistors to integrated circuits
- Established open hardware architecture (slots)

An Early PDP-8

PDP-11/70 Front Panel

The VAX-11/780

The Microcom[uter Era

- Integrated circuits allowed the creation of the first microprocessor, the Intel 4004, in 1971
- Microprocessors were first used for embedded controls, but still required separate memory, I/O, etc
- First microcomputer was the MITS Altair 8800, a DIY kit in 1974, using the Intel 8080
- Followed by the IMSAI-8080 and many others
- Followed by the Motorola 6502 based Apple I, then Apple II
- And then the IBM-PC in 1982

The Altair 8800

The famous A[[le][

The IBM-PC

The Market Mitosis – Part 1

Microcomputers

- Use microprocessors
- Designed for use by people
- General purpose
- Use packaged programs (applications)
- Will have human-friendly (somewhat) display and keyboard
- Are designed and priced for business and consumer use

Embedded Controllers

- Use microprocessors
- Are part of industrial machinery
- Specific purpose
- Run only a single program for a single task
- Minimal or no human interface
- Are designed and priced for industrial use

The Market Mitosis – Part 2

- If all the elements of a microcomputer are put on a single silicon chip, you have a System-on-a-Chip (SoC)
- SoC's were the logical development for embedded microprocessor systems
- But why not use an SoC to build something a human could interact with?
- If the SoC is configured for controlling electrical signals, we have a microcontroller (aka uC)
- If the SoC is configured for providing the functions of a traditional computer, we have a Single Board Computer (SBC)
- These two functions define a non-exclusive spectrum

Microcom[uter Evolution

- Moore's law is the observation that the number of transistors in a dense integrated circuit (IC) doubles about every two years. Moore's law is an observation and projection of a historical trend. Rather than a law of physics, it is an empirical relationship linked to gains from experience in production.
- From Gordon Moore, co-founder of Fairchild Semiconductor
- From the IBM-PC onwards, microcomputers continued to get *exponentially* smaller, faster, and much more powerful due to Moore’s law
- As processor power increased, Graphic User Interfaces (GUIs) became possible
- From desktops we went to laptops
- And from laptops we went to smart phones

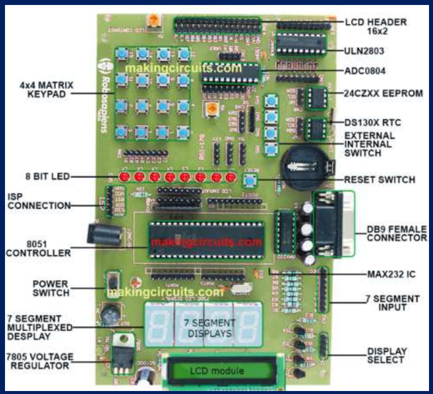
The Macintosh 128K

The all-in-one iMac

The iPhone 1

Microcontroller Evolution

- Originally microprocessors with separate memory, I/O
- As chip transistor density increased, it become possible to integrate all components onto an SoC
- SoCs evolved from embedded systems to user-accessible
- An SoC microcontroller made in a user-accessible format is called a microcontroller board or just “board” for short
- All boards offer General Purpose I/O (GPIO) pins which do a variety of things
- First and most famous user accessible microcontroller is the Arduino Uno
- We now have hundreds of microcontroller boards to choose from
- See https://makingcircuits.com/blog/types-of-microcontroller-boards-and-their-applications/ for details and examples



An embedded microcontroller

The Arduino Uno

The Make/Digi-Key Boards Guide

Formal End of Lesson 2

In next week's exciting episode

- Distribution of laptops (no, really!)
- Distribution and inventory of USB drives
- Installation of Arduino and Fritzing software
- Installation of ESP32 serial port drivers (Windows only)
- Electricity basics review (time permitting)