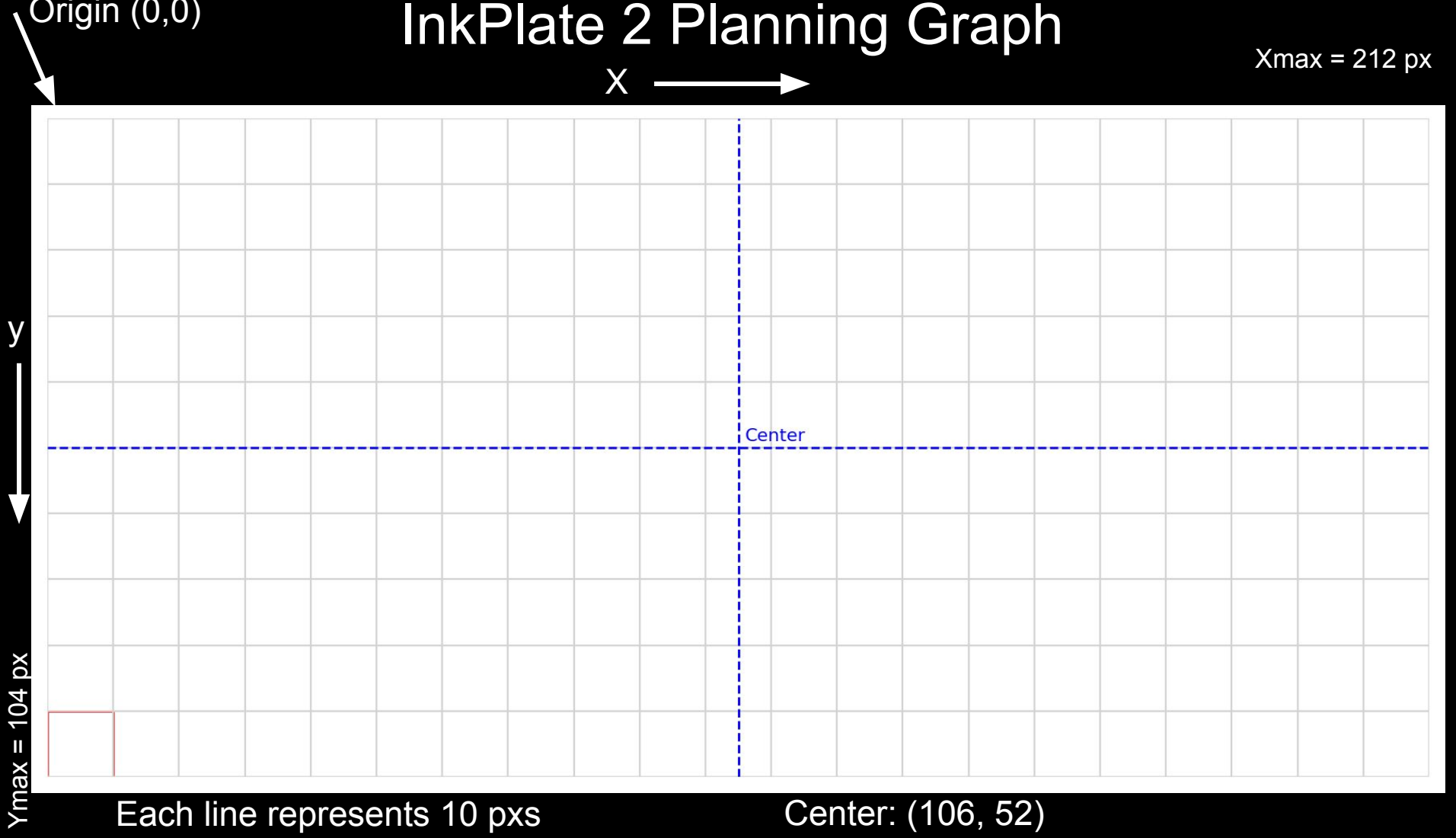
InkPlate 2 Planning Graph



Each line represents 10 pxs

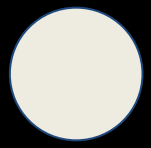
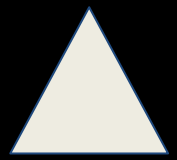
Center: (106, 52)

- setCursor(x,y) places the cursor at the top left most of your object
- InkPlate2 has 3 colors: 0 = white, 1 = black, 2 = red

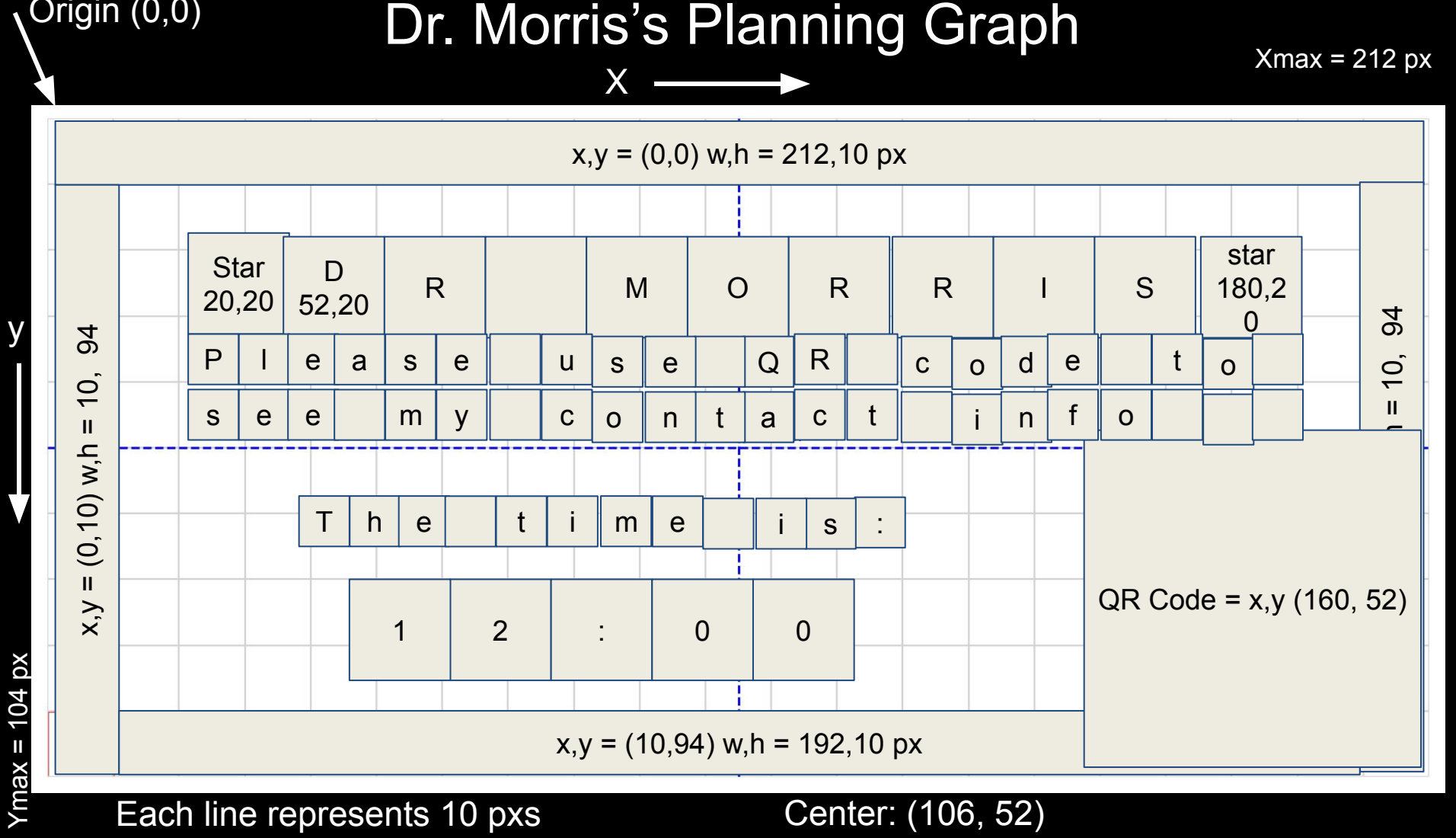
InkPlate 2 Planning Graph Parts Bin

Copy/Paste	Code	Notes
	display.drawImage(x, y, image, 16, 16, color)	16x16 px bitmap image
	setTextSize(1)	one text character = 8x8 px
	setTextSize(2)	one text character = 16x16 px
	fillRect(x, y, w, h, color)	draws a solid rectangle *adjustable*
	drawRect(x, y, w, h, color)	draws a hollow rectangle *adjustable*

InkPlate 2 Planning Graph Parts Bin

Copy/Paste	Code	Notes
	drawLine(x1, y1, x2, y2, color)	draws a straight line *adjustable*
	drawCircle(x, y, r, color)	draws a circle (x, y = center) *adjustable*
	setTextSize(2)	draws a triangle using 3 coordinate points *adjustable*
Add your own here!		
Add your own here!		

Dr. Morris's Planning Graph



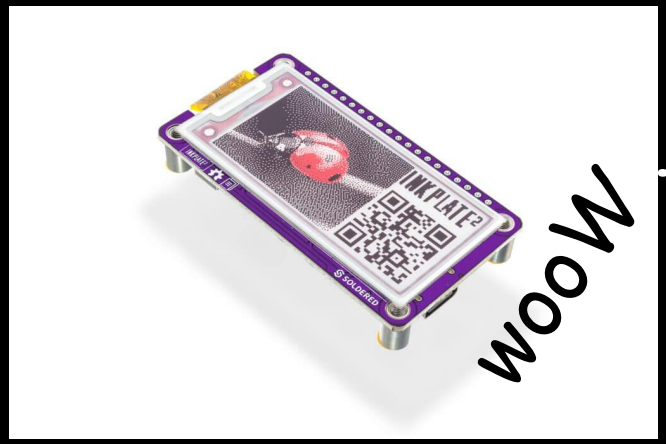
- setCursor(x,y) places the cursor at the top left most of your object
- InkPlate2 has 3 colors: 0 = white, 1 = black, 2 = red

InkPlate 2 Pod Unit

created by Dr. Harvey Morris
GT educator,
Carver Middle School
Tulsa Public Schools

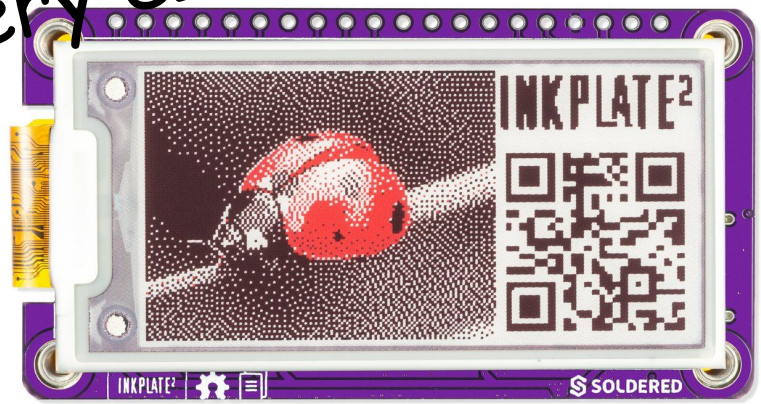
Questions or support?
morrishm@gmail.com

Welcome to the InkPlate 2 Pod!



woow.

very eInk



much time!

so weather!

Photo credit: e-radionica (www.e-radionica.com)

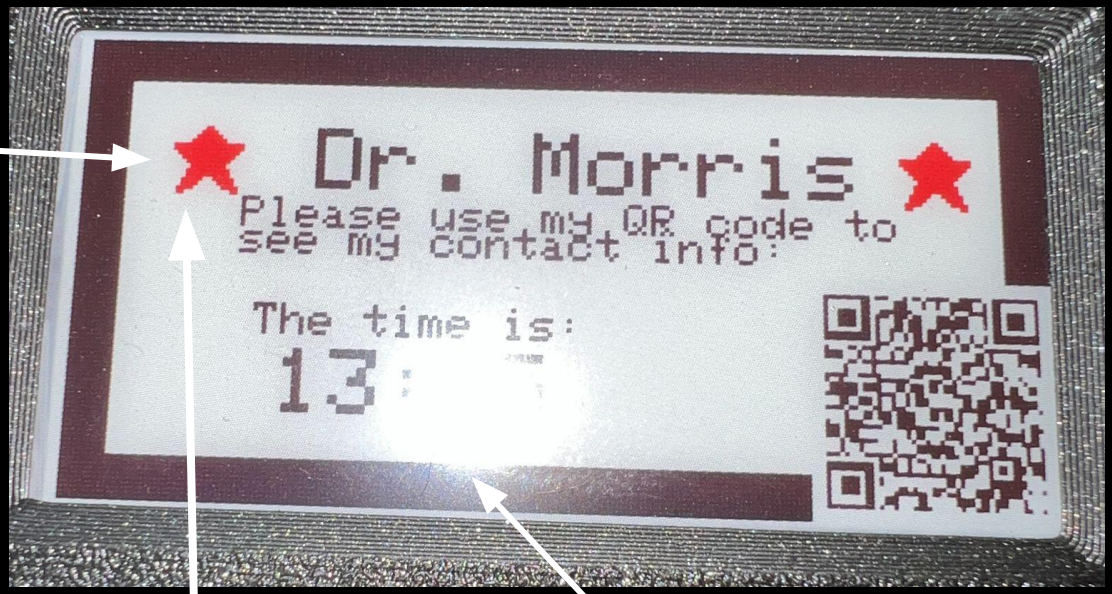
Icons from OpenMoji, licensed under CC BY-SA 4.0

Introduction Lesson: Objectives

- Used to introduce the pod to the class
- Build excitement and curiosity about the pod
- Introduce beginning computer and coding concepts
  - how computers “think”
  - binary vs. code
- Establish WHY we use programming languages
- Create a shared vocabulary before using real hardware
- Integrates NAGC standards
- **IMPORTANT: Set a tone of persistence, risk-taking, and independence**

The Ultimate Goal: A Digital Badge

My name and information



Imported QR code that links to my Github

bitmap

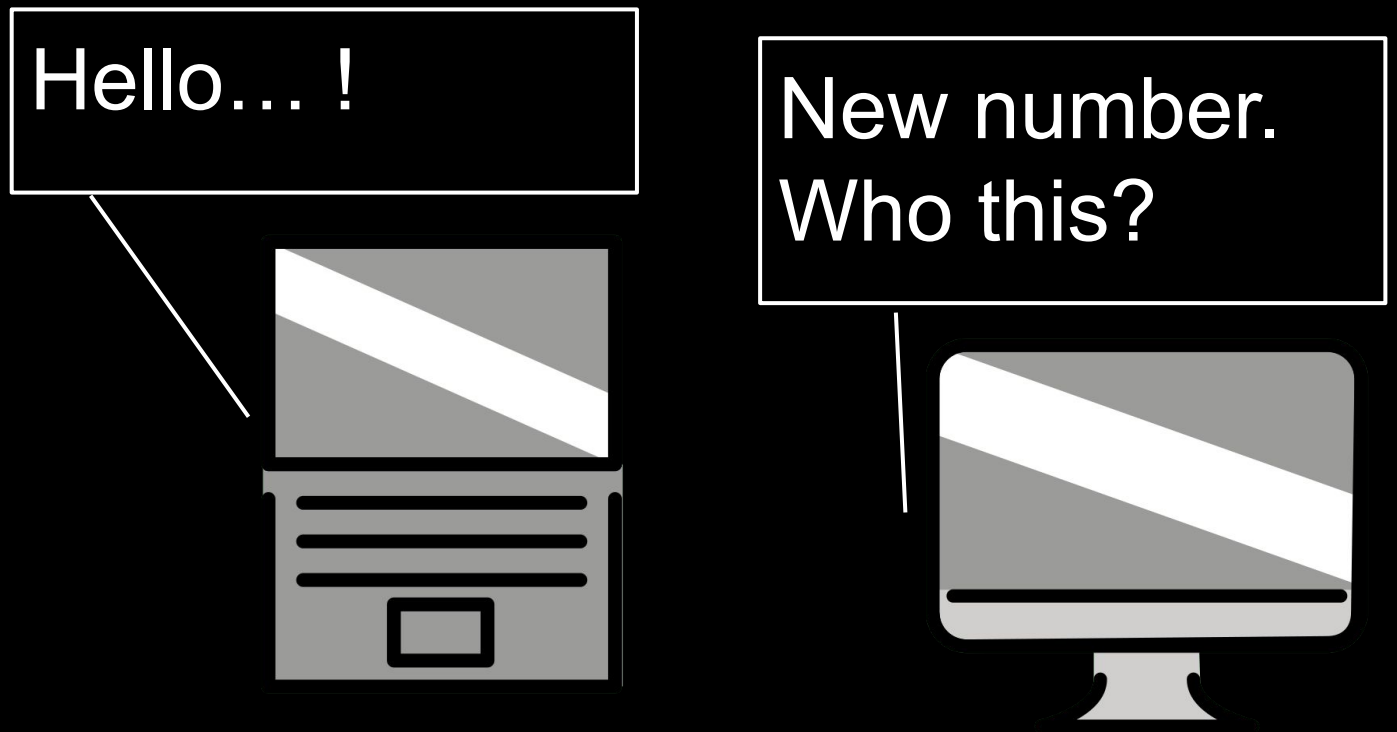
clock

Used as access to design the InkPlate 10 classroom display. Products will be shown during gallery walks and parent night.

Big Question

Do computers speak English when they talk to each other?

Think
Pair
Share
2 minutes



Machine Language

What we see…

What computer “sees”…

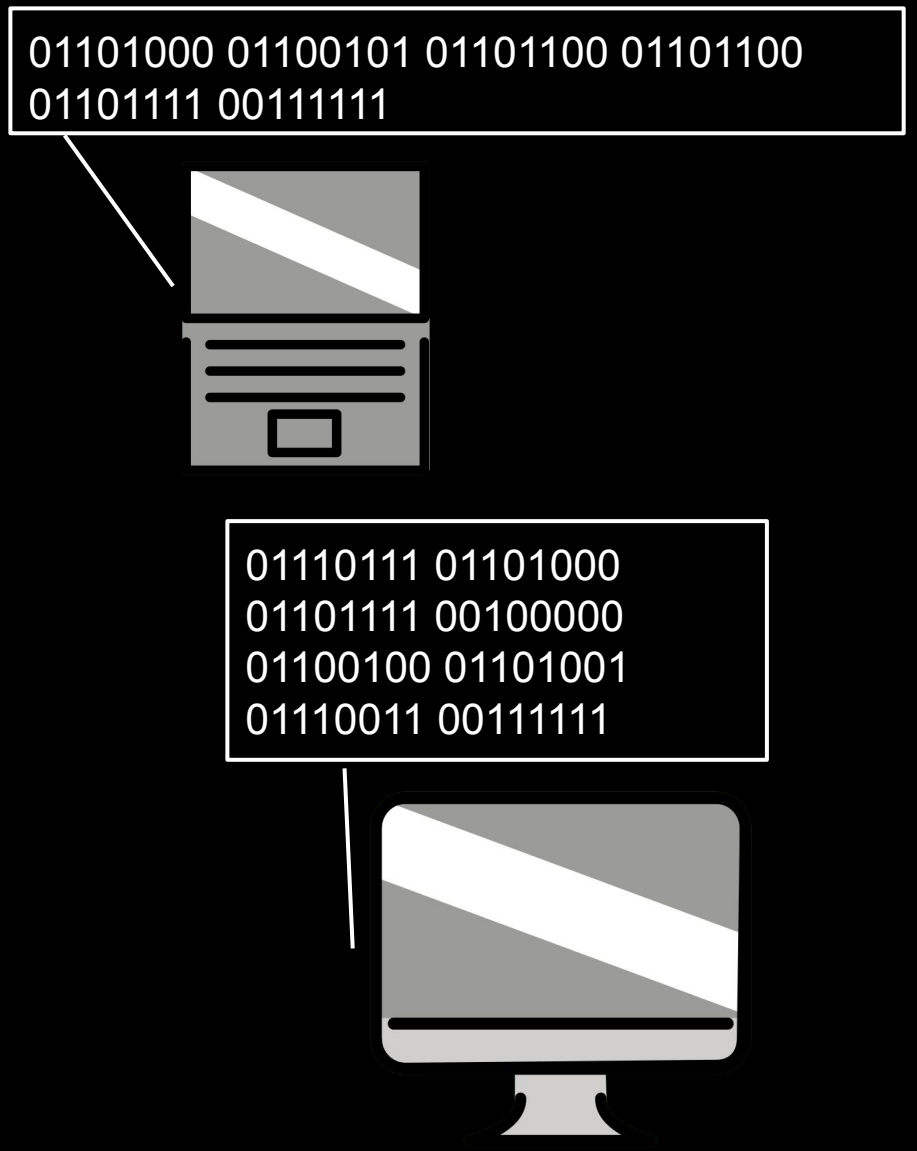
A 01000001

WHY?!

Each letter has a binary code – this is called ASCII

Machine Language

- Computers are built from circuits that can be on (1) or off (0)
  - This is called binary or machine language
- So everything has to be converted into binary
  - letters, numbers, pictures



Enter “Code”

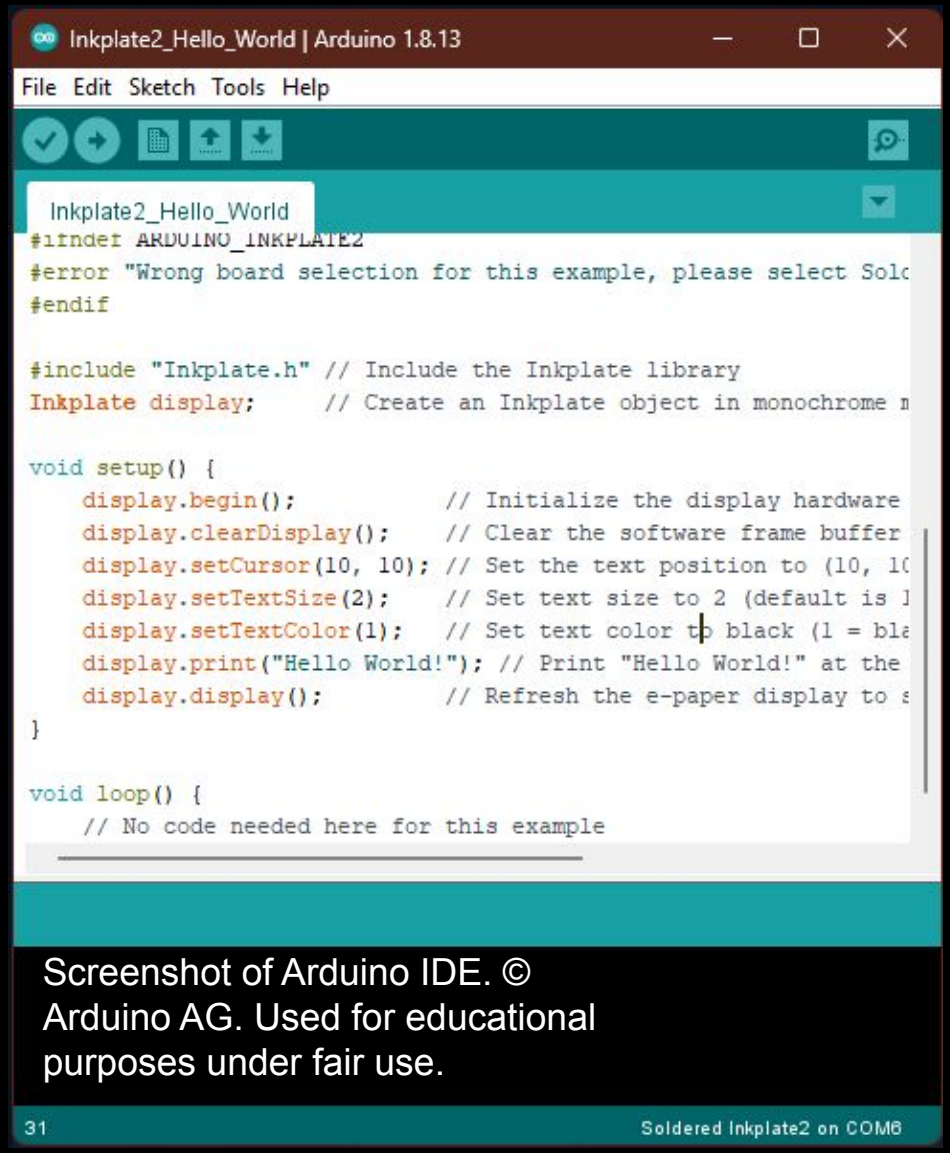
- Humans are really bad at reading binary.
- So, we invented “code” write machine language without having to read and write in binary.

What we write in code	What machine sees
display.print("Hi");	01001000 01101001

What is a Coding Language?

- Definition: Set of rules and instructions that tells a computer what to do.
- Different coding languages specialize in different tasks:
  - Python - data, AI, beginner projects
  - Javascript - Powers websites and interactive elements
  - Swift - used to build iPhone and iPad apps
  - **C/C++ - powers machines, hardware, and fast code (like the InkPlate 2!)**
  - Lua - used for scripting in video games
  - HTML/CSS - designs the structure and style of web pages.

Arduino (A Variant of C++)



Screenshot of Arduino IDE. © Arduino AG. Used for educational purposes under fair use.

Arduino	🧠 Python / Scratch / Others
Controls **physical things**	Controls **virtual things**
Ex: screens, sensors, motors	Ex: games, apps, websites
Runs on a **microcontroller**	Runs on a **computer**
Needs **electrical components**	Needs **code only**
Great for **hardware projects**	Great for **logic & creativity**

We will code in Arduino IDE!

Machine Language

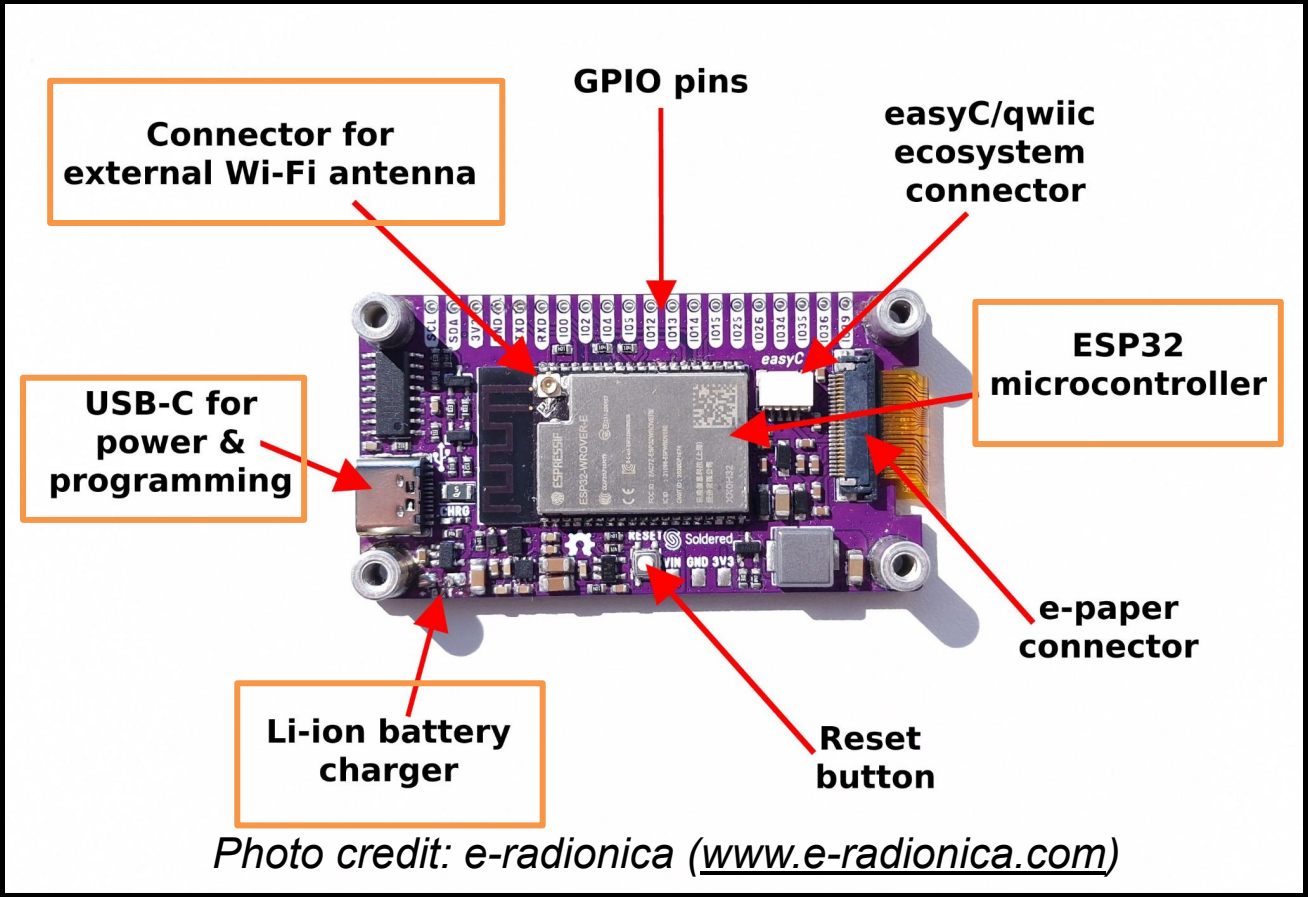
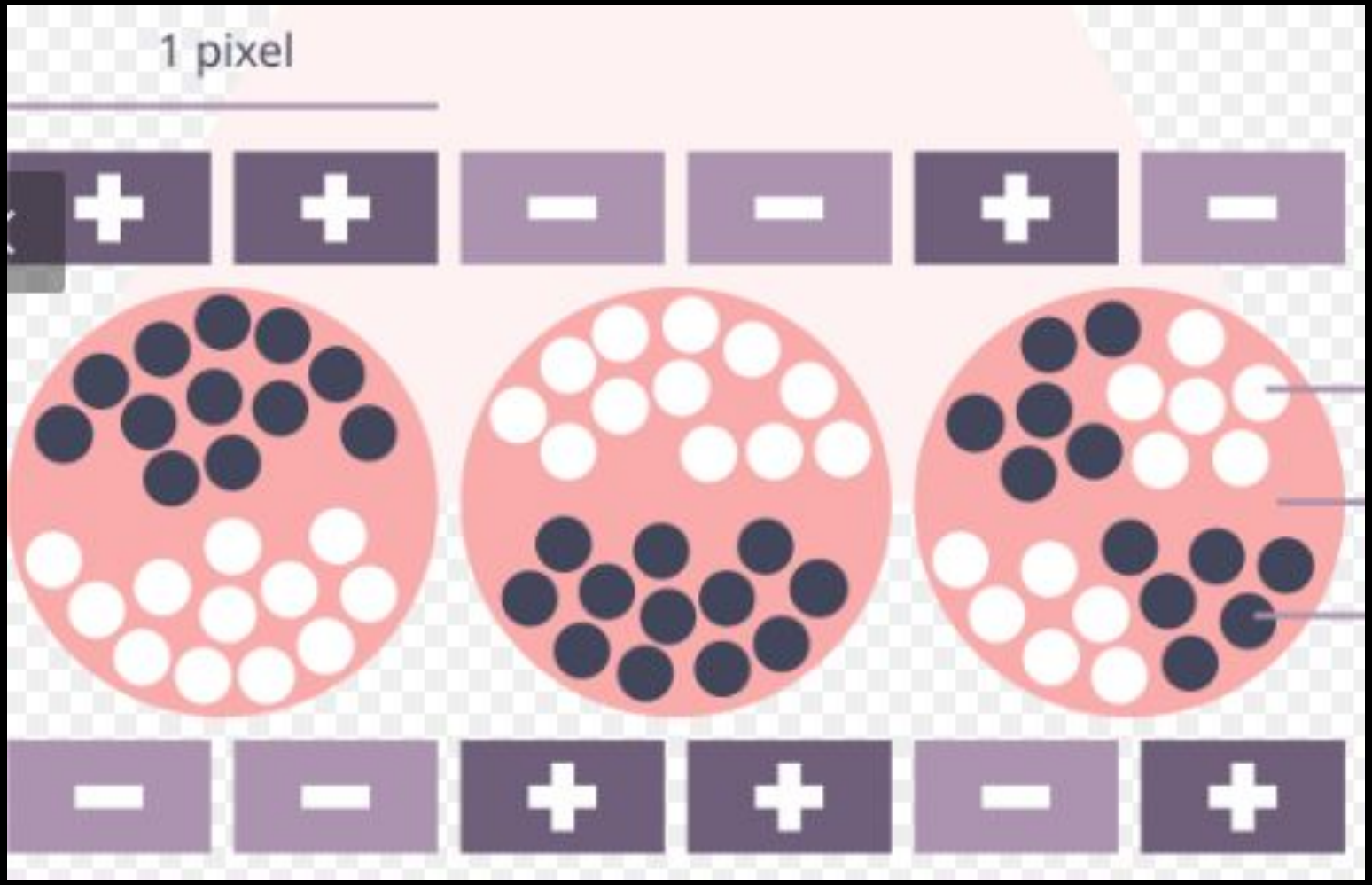


Photo credit: e-radionica (www.e-radionica.com)

Flow in information:

Computer ➡ ESP32 micro ➡ eInk display

eInk Pixels = + and - Charged Ink



Term	Definition
Binary	A system of 1s and 0s that computers use to represent information
Machine Language	The lowest-level programming language, directly understood by computers
ASCII	A standardized code that assigns binary values to letters, numbers, and symbols
Coding Language	A set of rules for writing instructions a computer can understand and run
Arduino	A beginner-friendly coding platform based on C++, used to program microcontrollers
C++	A powerful coding language used in robotics, machines, and hardware
ESP32	The microcontroller chip inside the Inkplate 2 that runs your code
eInk Display	A screen that mimics paper using black, white, and red charged ink particles
IDE	Integrated Development Environment – the software where you write code (e.g., Arduino IDE)
setCursor(x, y)	A command that tells the screen where to begin drawing text or shapes

Alignment with National Association of Gifted Children (NAGC) Standards

NAGC	Description	Lesson Example	Aligned
1.1.1	Students demonstrate growth in multiple content areas through challenging learning opportunities.	*Intro to Inkplate + "What Is Code?"*	Foundational lecture series introduces students to e-ink, binary, and hardware logic, pushing beyond grade-level science and tech standards.
1.1.3	Students identify and solve authentic problems.	*Custom Layout Badge Project*	Students design a digital ID using coordinates and logic. They troubleshoot real layout and display constraints (e.g., pixel limits).
1.2.2	Students use technology to solve real-world problems.	*QR Code Badge with Time Display*	Students upload a QR image and NTP-connected clock sketch to the Inkplate 2, integrating internet-connected hardware design.
2.2.1	Educators use differentiated instructional practices to address the learning needs of gifted students.	*Bitmap Redesign Challenge*	Students critique pixel art designs (like the star) and improve them, offering multiple levels of visual and technical entry.
2.4.2	Educators model use of advanced technologies.	*All Wi-Fi Enabled Lessons*	Lessons explicitly model WiFi setup, image rendering via URL, and hardware-software coordination.

Alignment with National Association of Gifted Children (NAGC) Standards

NAGC	Description	Lesson Example	Aligned
3.1.1	Students develop perseverance and resilience when faced with challenge.	*Debugging QR + Clock Display*	Students encounter and overcome issues (like screen cutoff, wrong time zones), fostering grit and logical reasoning.
3.2.1	Students explore independently and develop self-direction.	*Extension: Personalized GUI Project*	Students create a layout from scratch, choosing text, images, and functions. No template, high autonomy.
4.1.1	Learning environments are intellectually and emotionally safe.	*Lesson 0: What Is a Display?*	Sets norms for experimentation, making mistakes, and exploring unfamiliar hardware in a safe GT environment.