

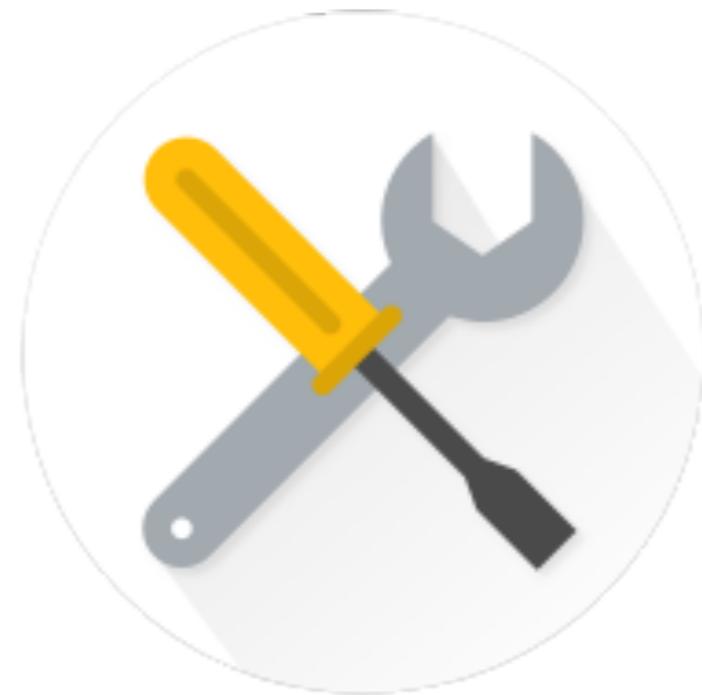
Lecture #1

Introduction to IoT

Android Things 2019

Prerequisites

- Modern programming language
- Object oriented
- Statically types
- IDE - IntelliJ/Android Studio



What you should know...

- Basics:
 - Object-oriented programming
 - Classes, methods
 - Exception handling
 - Android



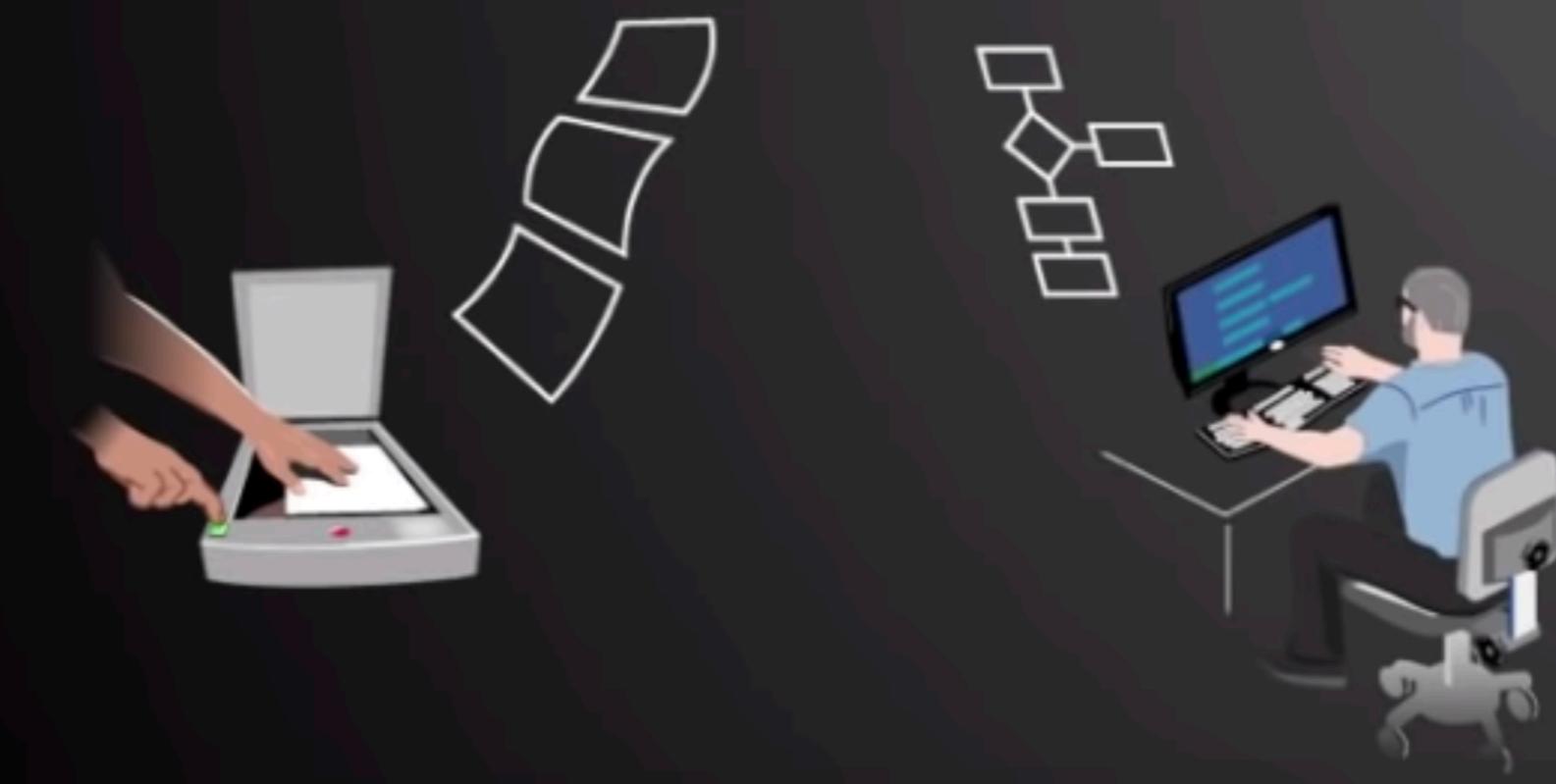
Bonus

- Functional Programming
- Lambdas
- Higher Order Functions
- Reactive Programming
- Kotlin

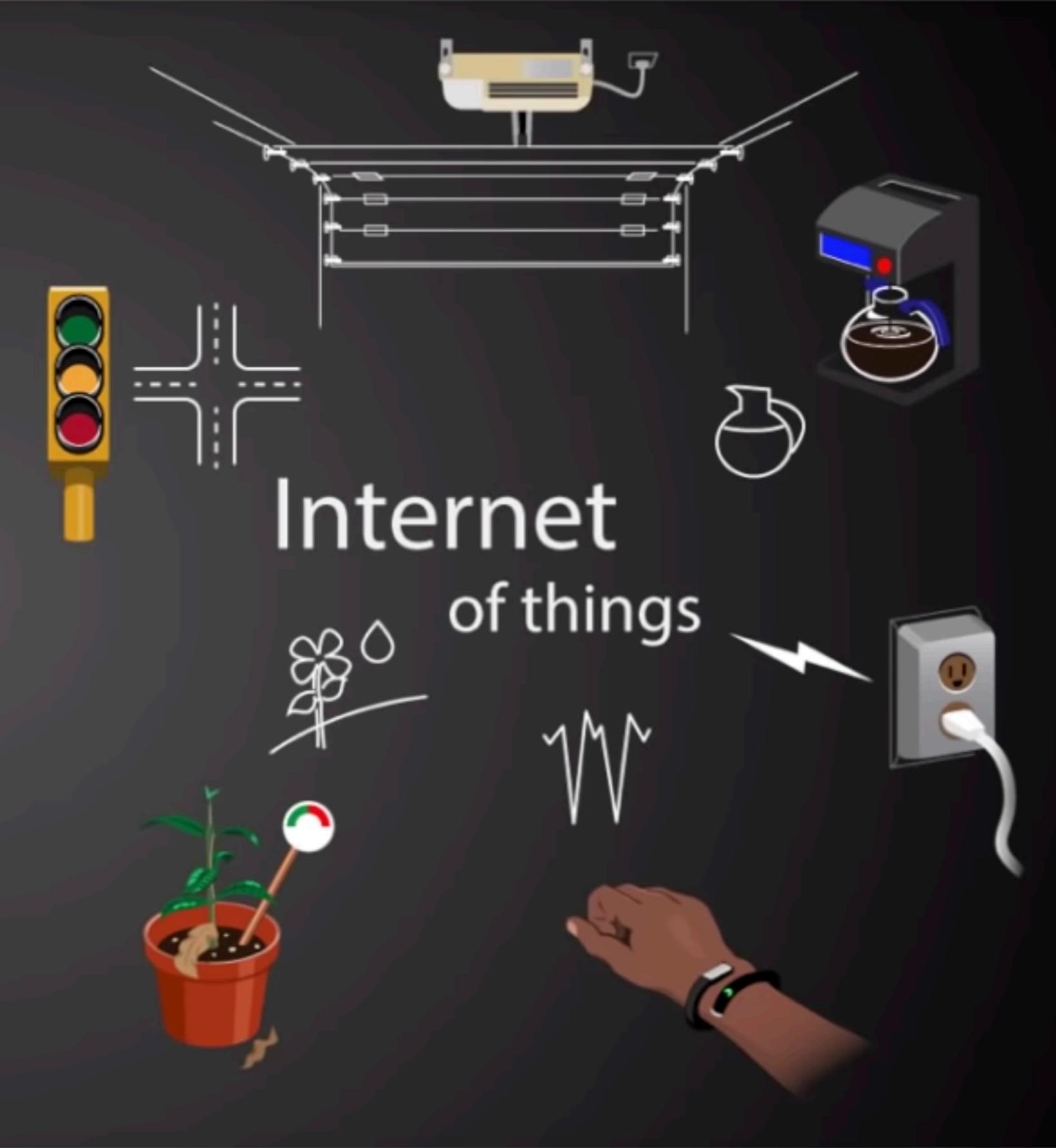




Internet of people



Internet of things



Why

- Collect and aggregate data.
- Remote control devices.
- Automate certain tasks.



Image source: <https://newsignature.com/articles/iot-now-eric-stein-new-signature-solutions-architect/>

What

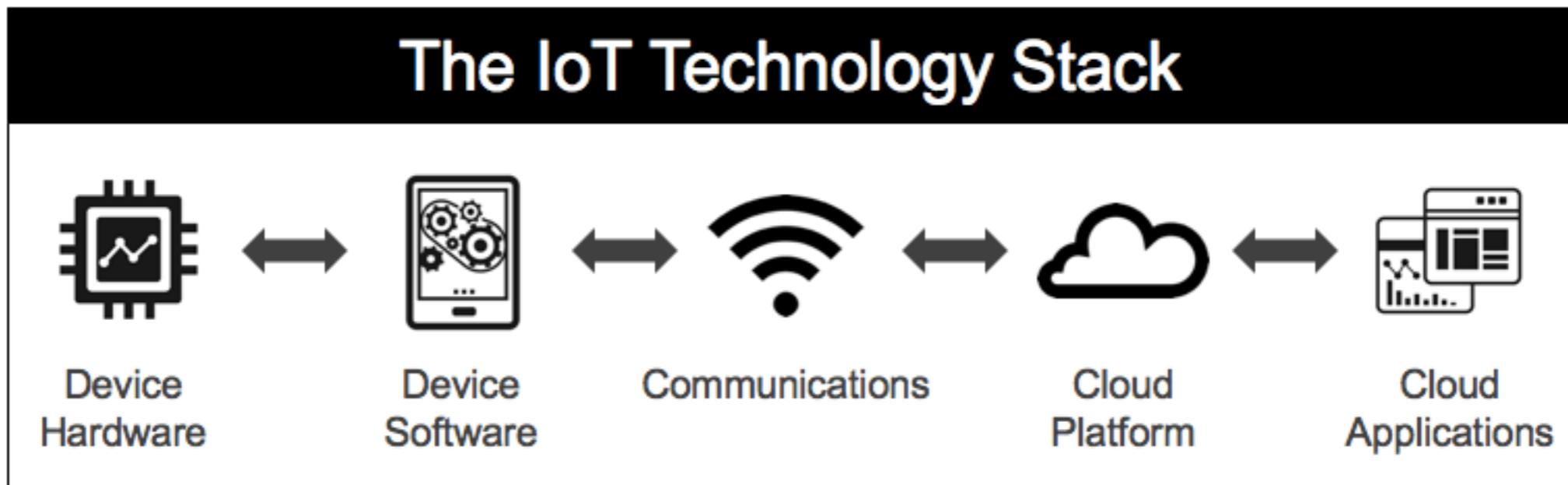
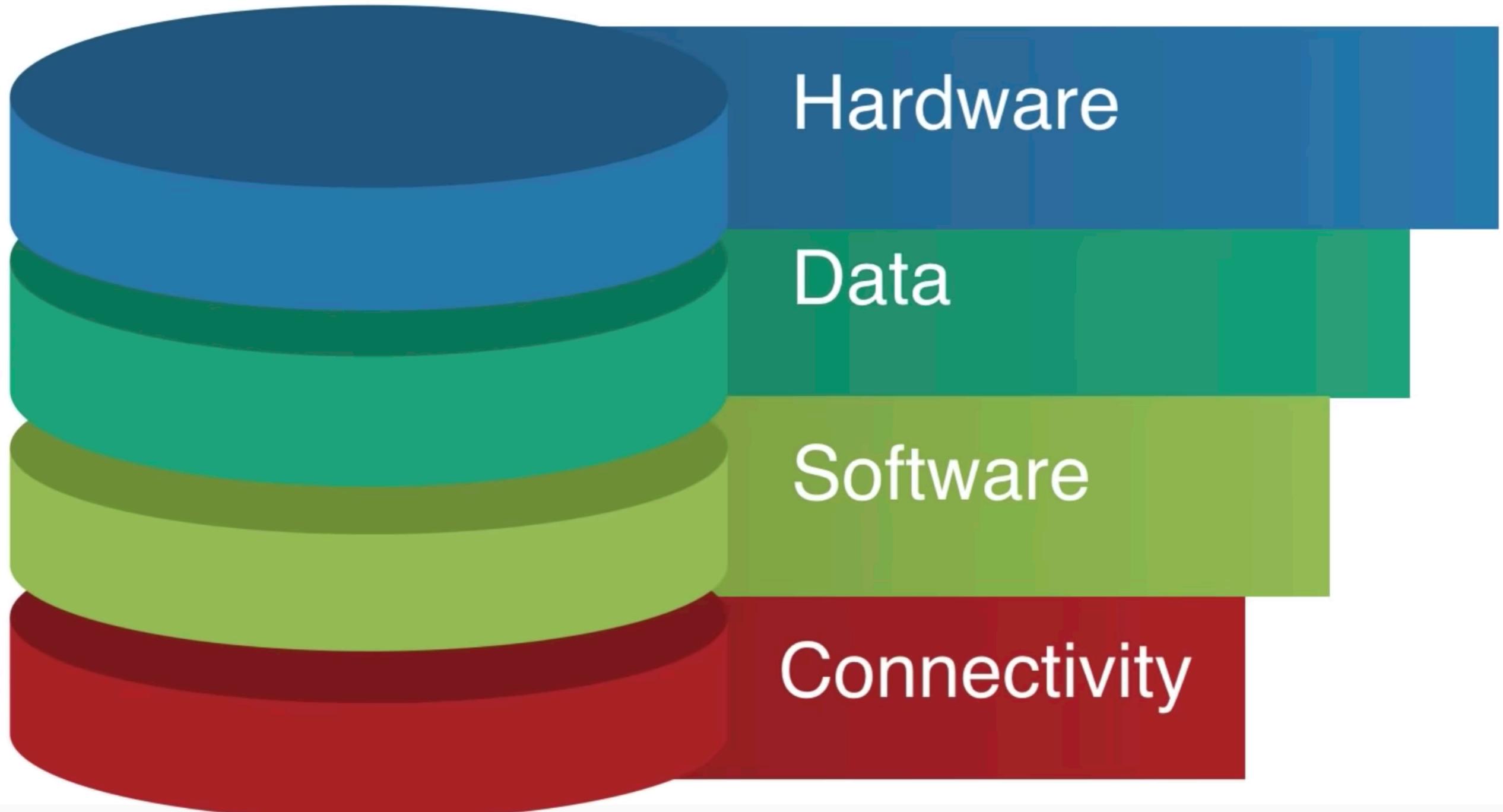
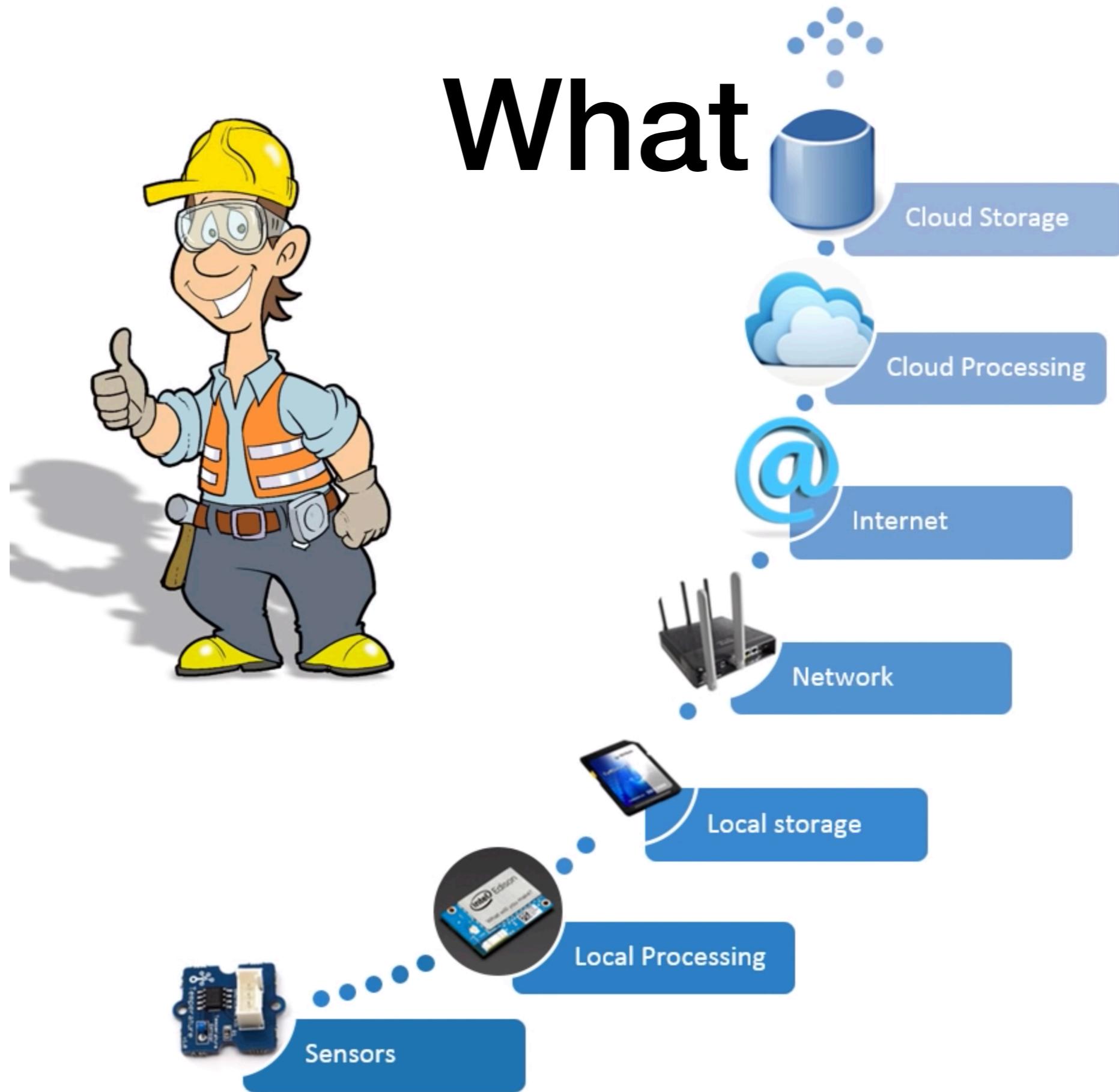


Image source: <https://www.iotforall.com/iot-product-managers-guide-iot-technology-stack/>

What





What

Internet of Things



What

Internet of Things



What

Internet of Things



A Google Home smart speaker is the central focus, positioned on a light-colored wooden coffee table. In the background, a person wearing a green t-shirt is standing near a sofa. A red balloon is visible on the right side of the frame.

Introducing
Google Home

<https://www.youtube.com/watch?v=KECfgrw02fU>

A Google Home smart speaker is the central focus, positioned on a light-colored wooden coffee table. In the background, a person wearing a green t-shirt is standing near a sofa. A red balloon is visible on the right side of the frame.

Introducing
Google Home

<https://www.youtube.com/watch?v=KECfgrw02fU>



<https://www.ratp.fr/en/groupe-ratp/engineering/fully-automated-century-old-metro-lines>



Mission

Technology

Journey

What's Next

Early Riders

We're building the world's
Most Experienced Driver™



PLAY



<https://waymo.com/>

**"The Internet of everything will have five to 10 times
the impact on society as the Internet itself"**

Cisco CEO John Chambers

**"The Internet of everything will have five to 10 times
the impact on society as the Internet itself"**

Cisco CEO John Chambers

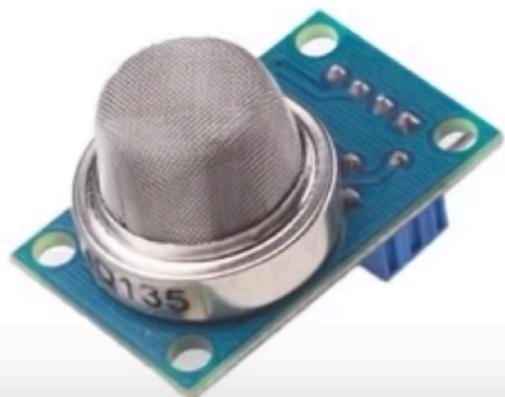
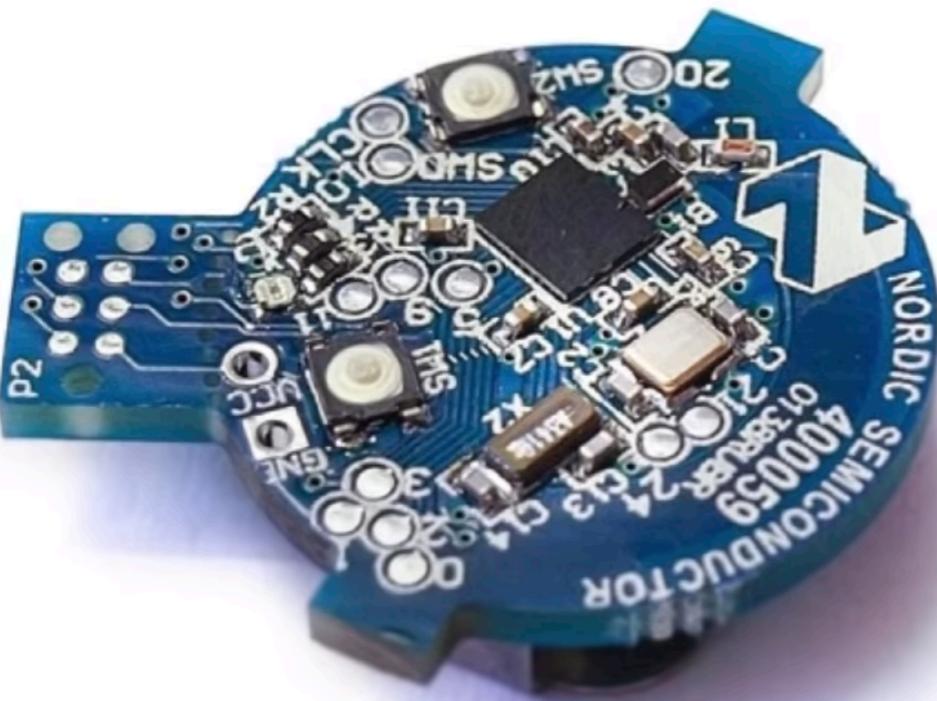
\$19 trillion in economic benefit and value over the next decade.

Sensors

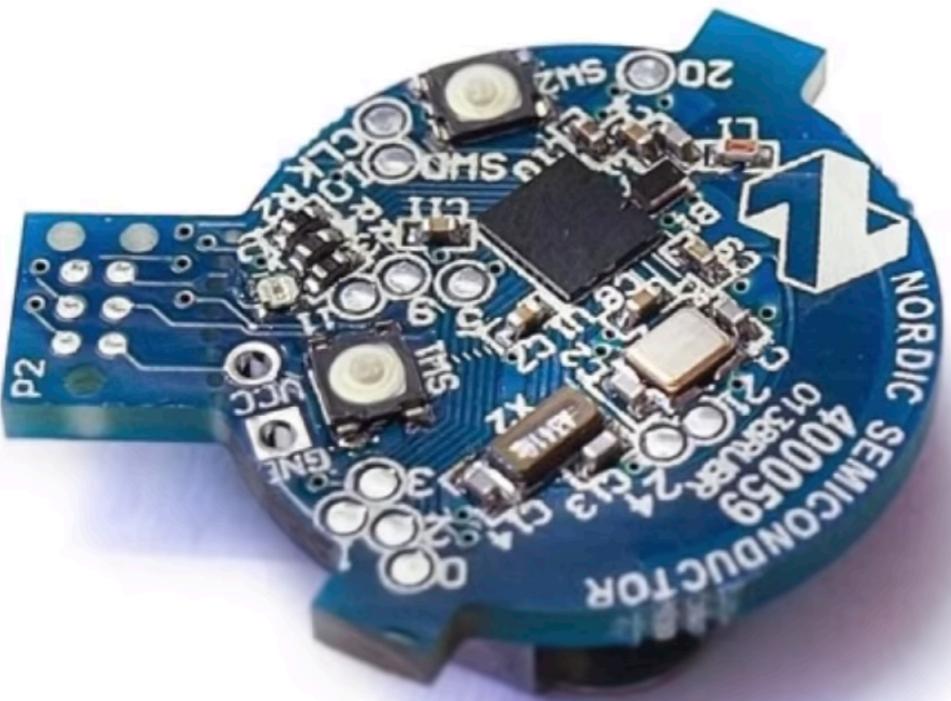
- Measure values.
- Send raw data.
- Low power.
- Almost no maintenance needed.



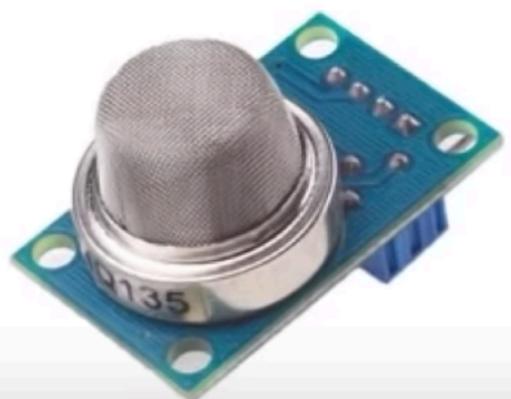
Sensors



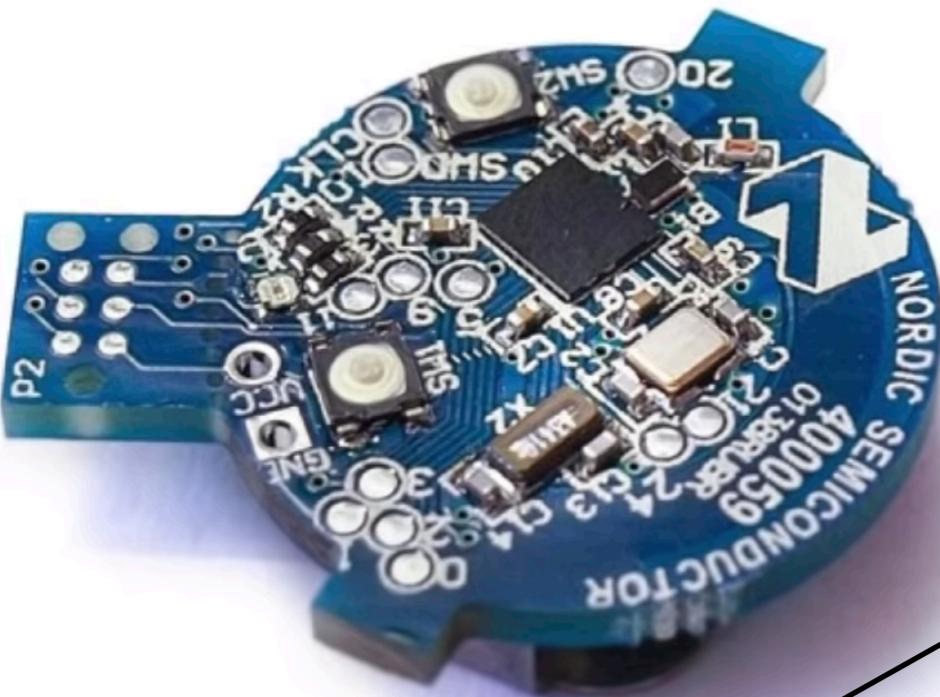
Sensors



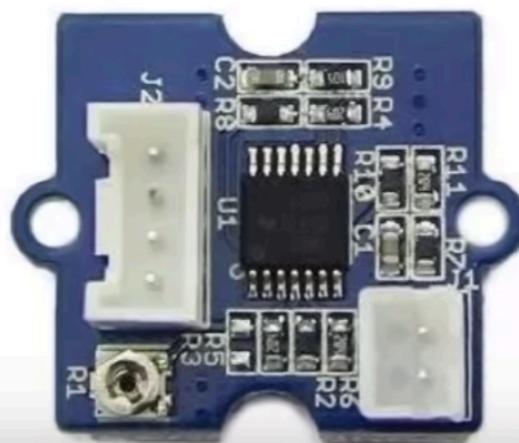
Bluetooth Beacon



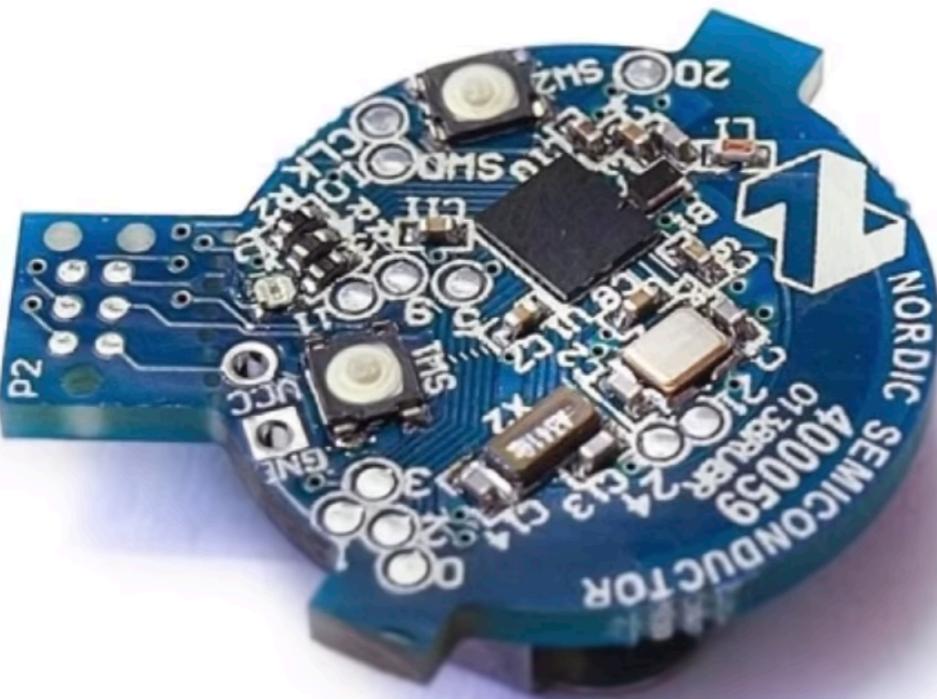
Sensors



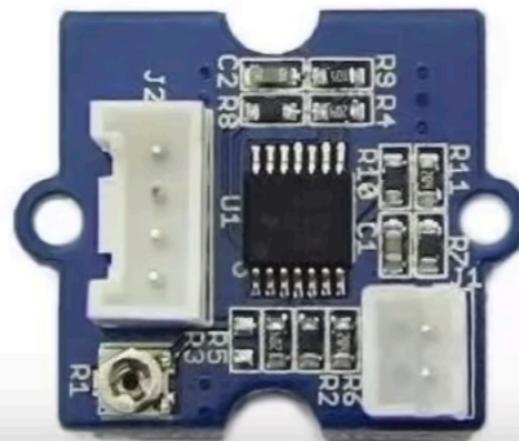
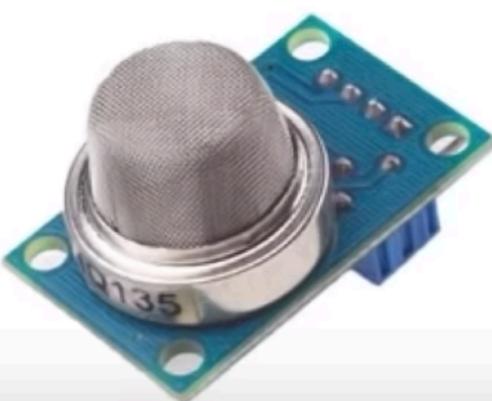
Air Quality Sensor



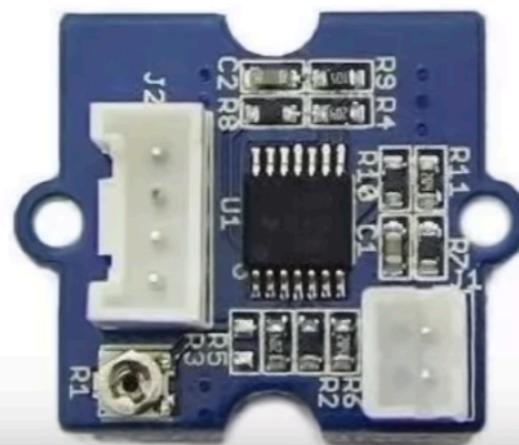
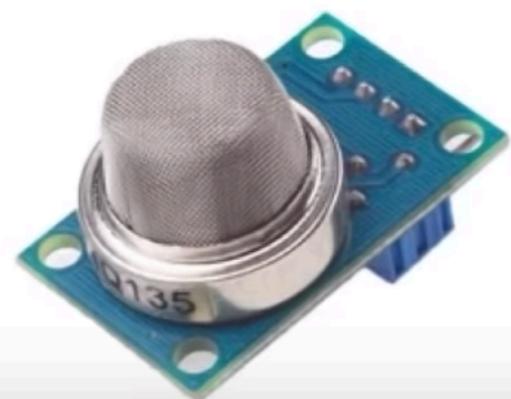
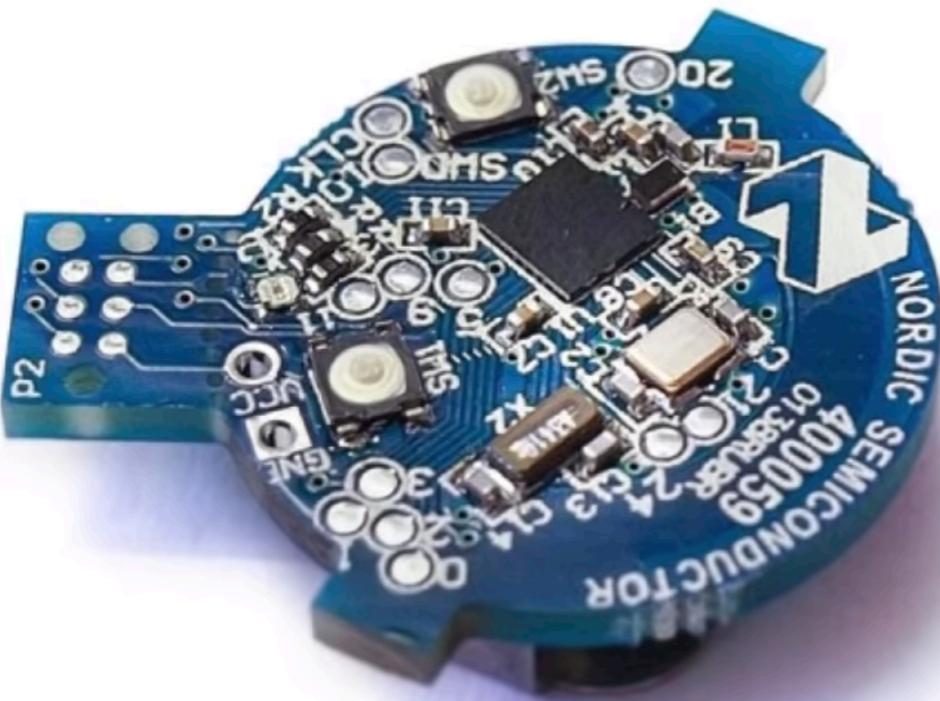
Sensors



**Galvanic Skin
Response Sensor**



Sensors



Consumer Products

Local Processing

- Collects sensor data.
- Light processing.
- Uploads data to the cloud.

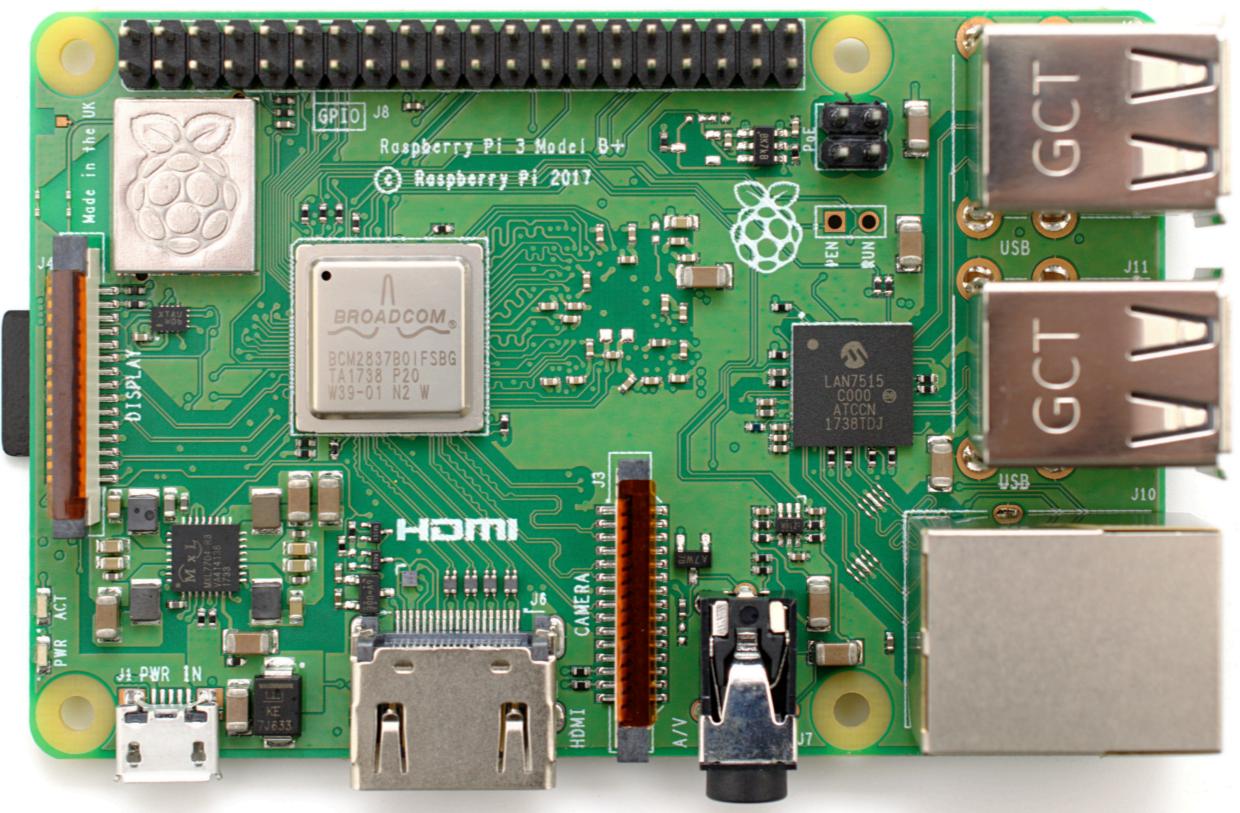


Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>

Local Processing

- Collects sensor data.
- Light processing.
- Uploads data to the cloud.

Edge/Fog Computing

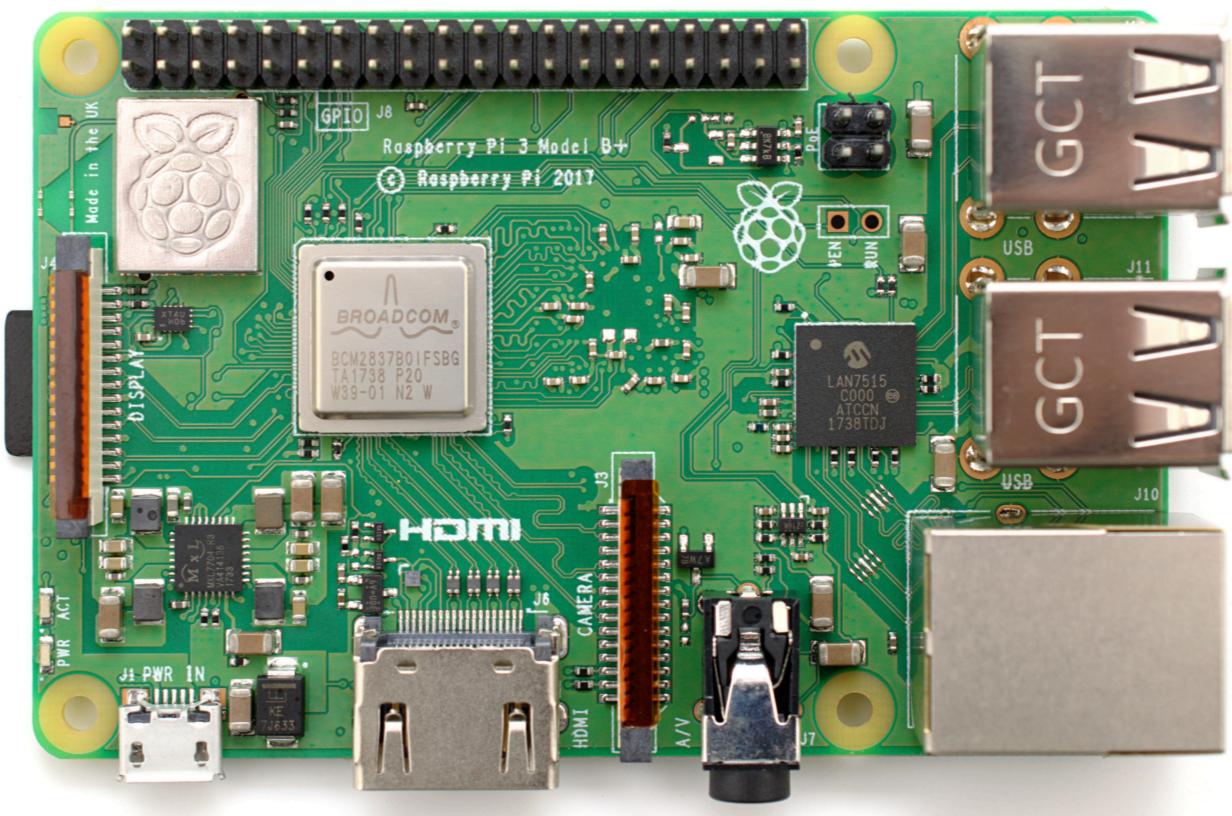
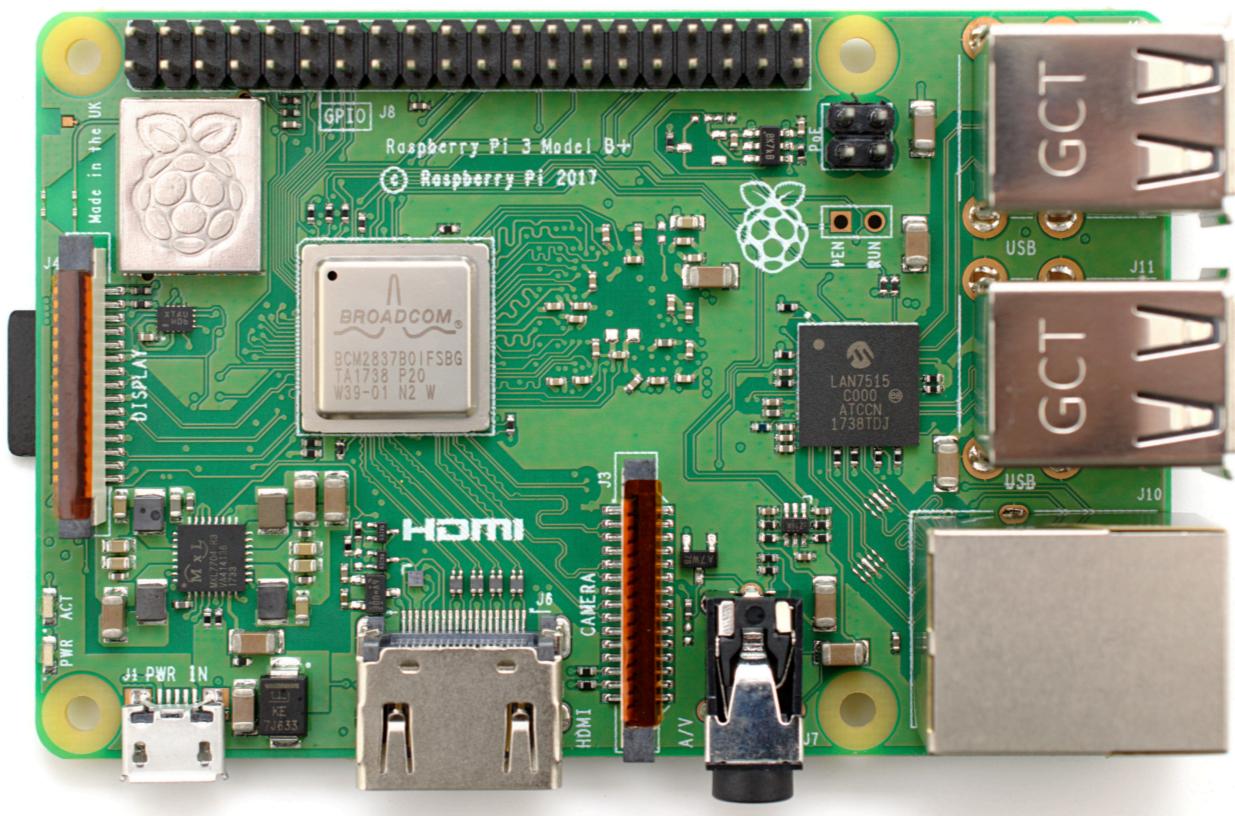


Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>

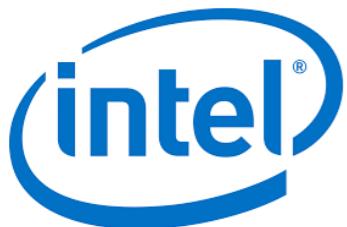
Local Processing

- Collects sensor data.
- Light processing.
- Uploads data to the cloud.



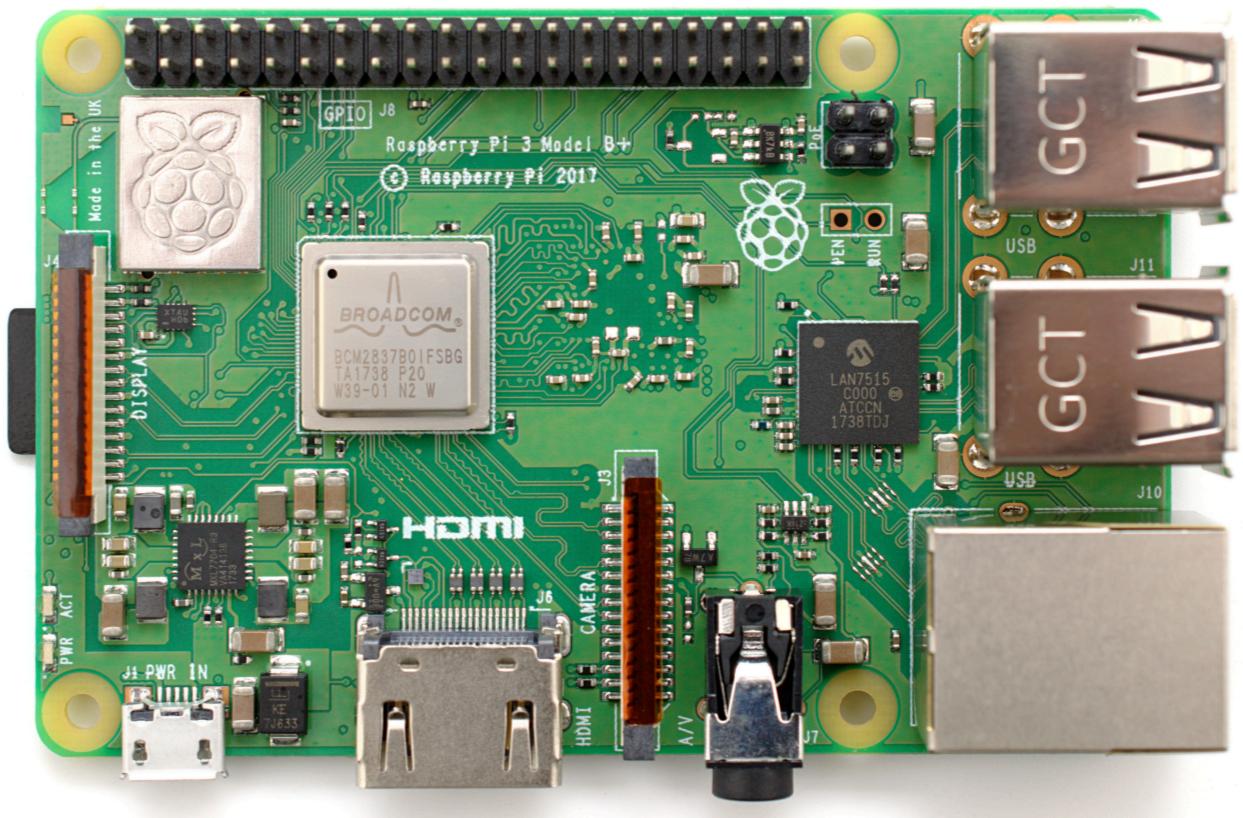
Edge/Fog Computing

Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



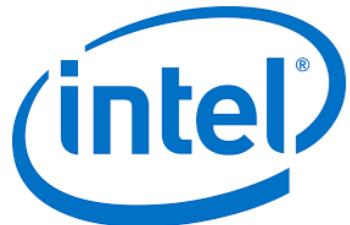
Local Processing

- Collects sensor data.
- Light processing.
- Uploads data to the cloud.



Edge/Fog Computing

Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



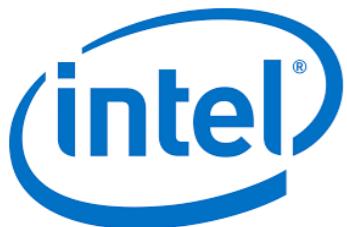
Local Processing

- Collects sensor data.
- Light processing.
- Uploads data to the cloud.



Edge/Fog Computing

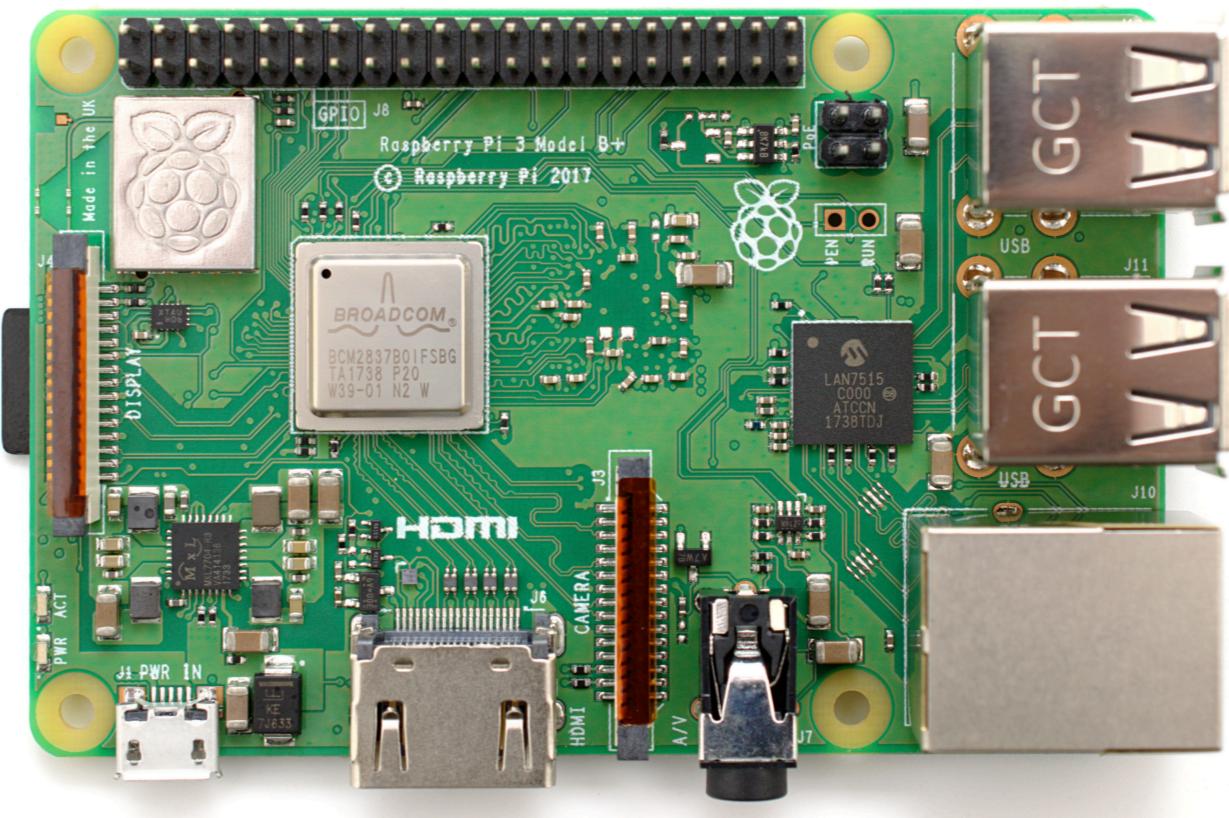
Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



Local Processing Persistence

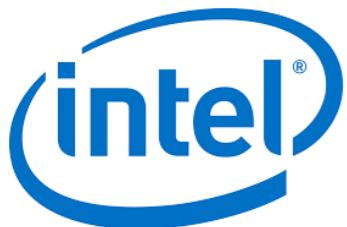


- Collects sensor data.
- Light processing.
- Uploads data to the cloud.



Edge/Fog Computing

Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



Network & Internet

- IoT Gateways.
- Connects multiple sensors and local processing units.
- Protocols:
 - CoAP
 - MQTT
 - HTTP
 - XMPP



Image source: <https://commons.wikimedia.org/wiki/File:Linksys-Wireless-G-Router.jpg>

Network & Internet

- IoT Gateways.
- Connects multiple sensors and local processing units.
- Protocols:
 - CoAP <http://coap.technology/>
 - MQTT <http://mqtt.org/>
 - HTTP
 - XMPP



Image source: <https://commons.wikimedia.org/wiki/File:Linksys-Wireless-G-Router.jpg>

Network & Internet

- IoT Gateways.
- Connects multiple sensors and local processing units.
- Protocols:
 - CoAP <http://coap.technology/>
 - MQTT <http://mqtt.org/>
 - HTTP <https://www.w3.org/Protocols/>
 - XMPP



Image source: <https://commons.wikimedia.org/wiki/File:Linksys-Wireless-G-Router.jpg>

Network & Internet

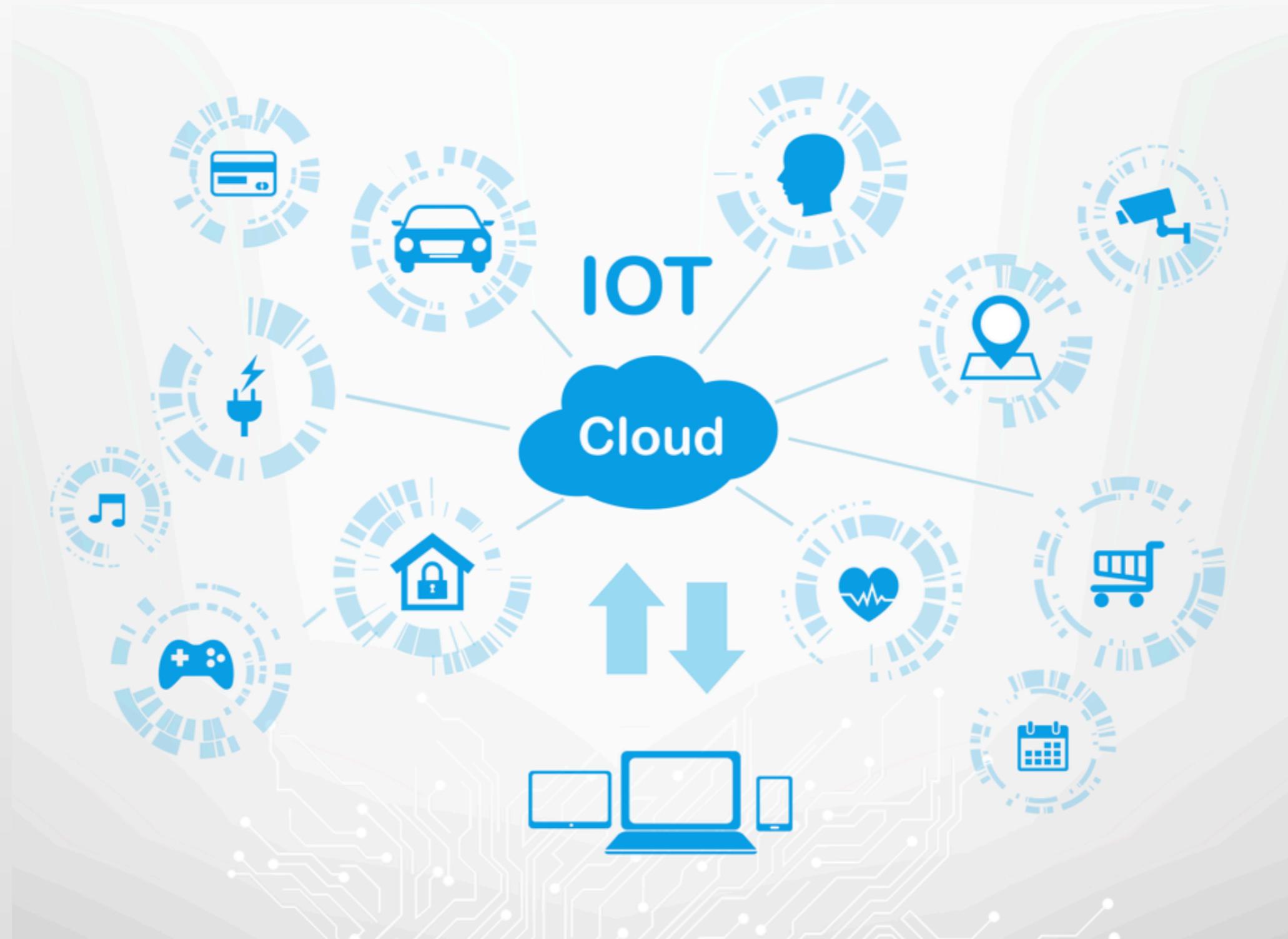
- IoT Gateways.
- Connects multiple sensors and local processing units.
- Protocols:
 - CoAP <http://coap.technology/>
 - MQTT <http://mqtt.org/>
 - HTTP <https://www.w3.org/Protocols/>
 - XMPP <https://xmpp.org>



Image source: <https://commons.wikimedia.org/wiki/File:Linksys-Wireless-G-Router.jpg>

Cloud Processing & Storage

- Aggregate.
- Store.
- Analyze.
- Predict.



How it started

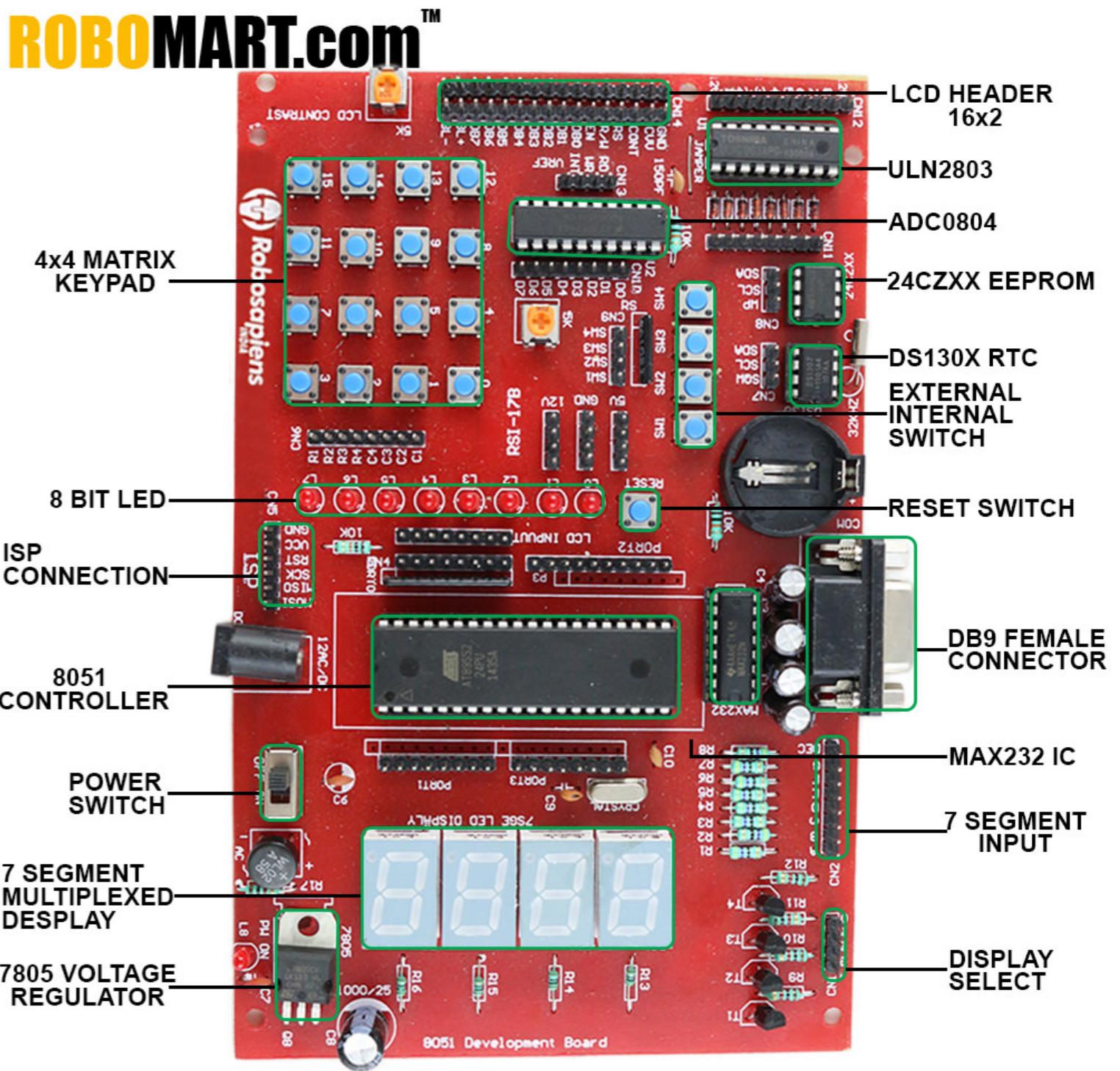
- August 26th, 1997
 - Bruce Perens
 - <https://lists.debian.org/debian-announce/1997/msg00026.html>



Image source: <https://www.oshwa.org/open-source-hardware-logo/>

Microcontroller

- Small programmable devices.
- Easily connectable.



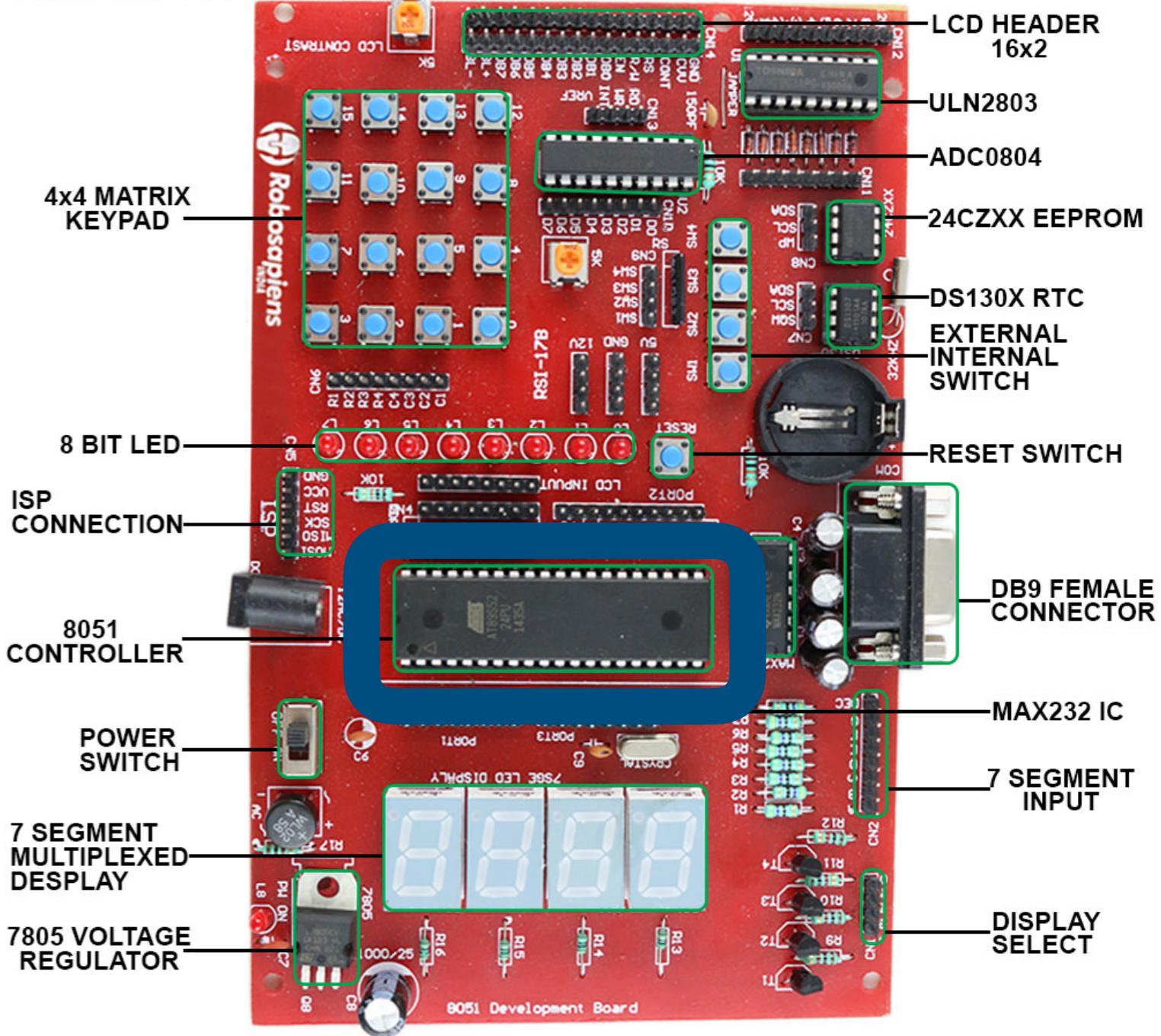
NOTE:THE PRODUCT MAY BE DIFFER FROM IMAGE GIVEN ABOVE

Copyrights by Robomart.com

Microcontroller

ROBOMART.com™

- Small programmable devices.
 - Easily connectable.



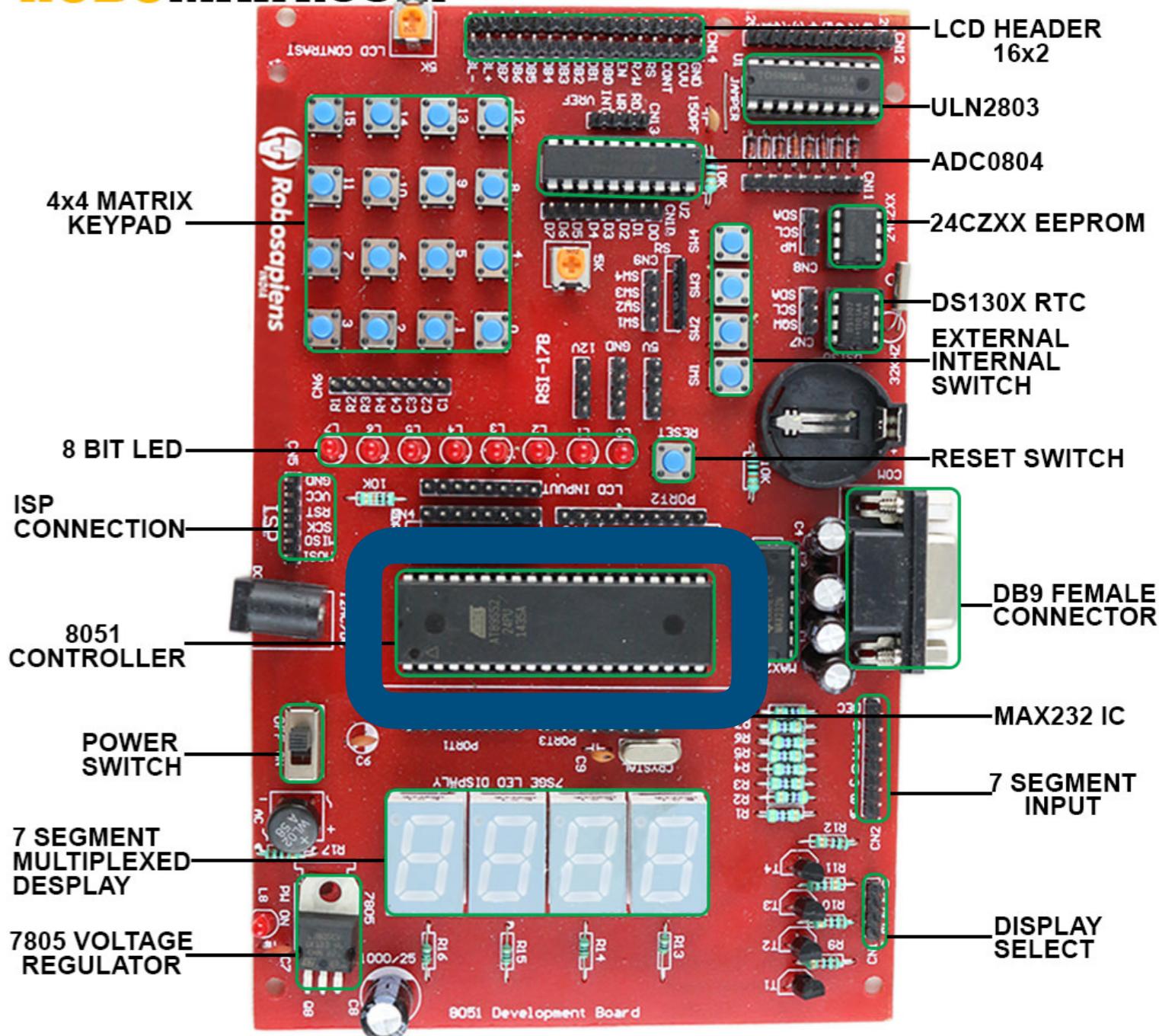
NOTE:THE PRODUCT MAY BE DIFFER FROM IMAGE GIVEN ABOVE

Copyrights by Robomart.com

Microcontroller

ROBOMART.com™

- Small programmable devices.
- Easily connectable.





1972

Microcontroller

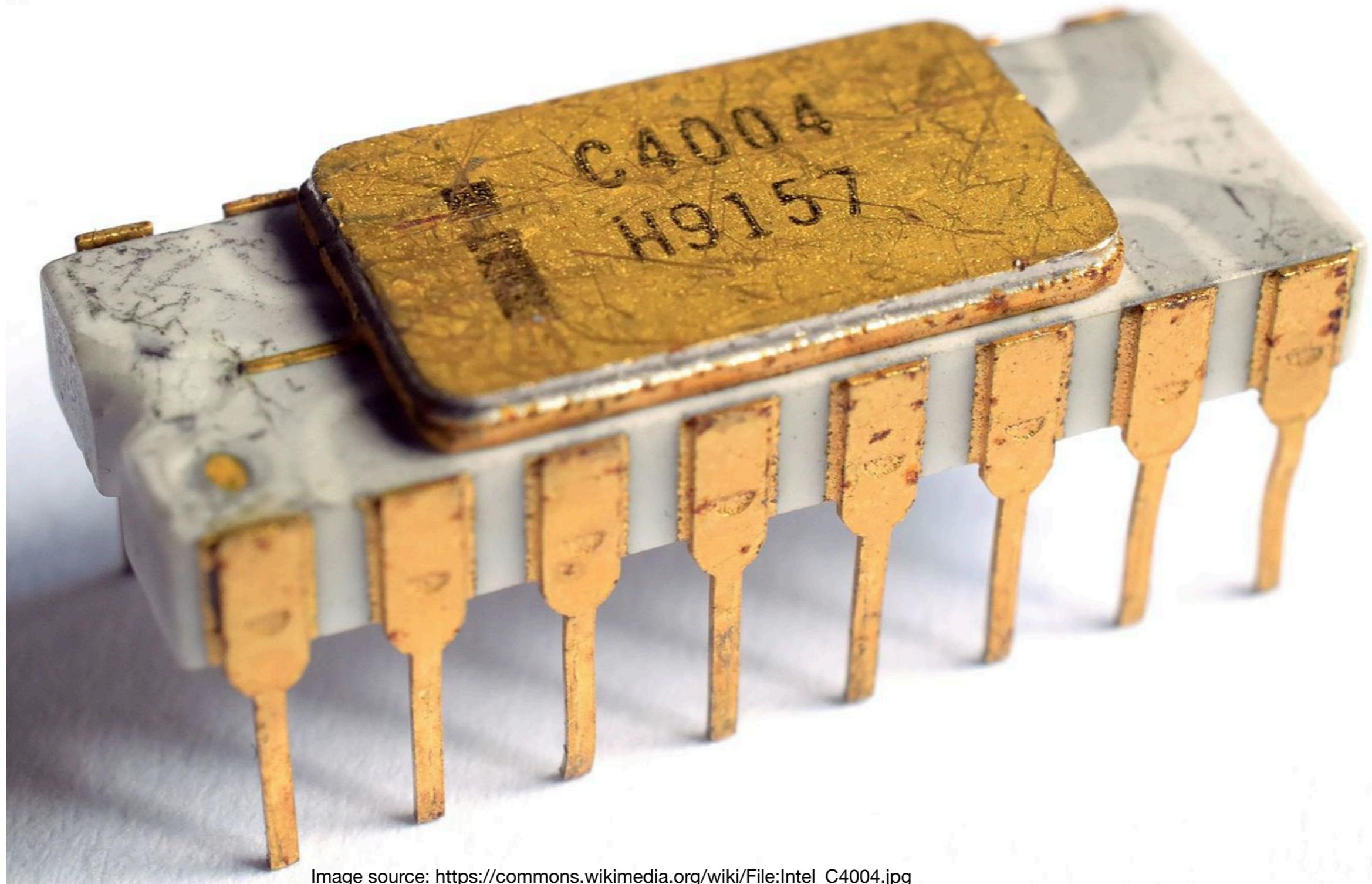


Image source: https://commons.wikimedia.org/wiki/File:Intel_C4004.jpg

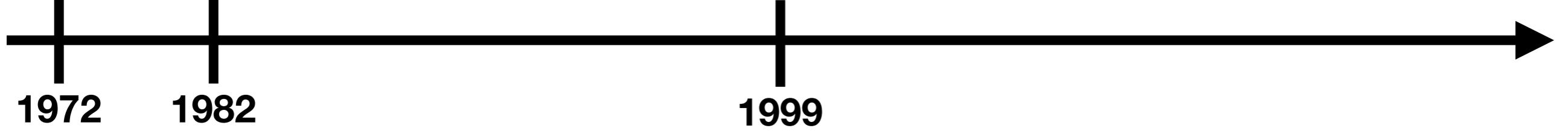


First Internet Connected Device



CMU's connected Coke machine

https://www.cs.cmu.edu/~coke/history_long.txt



Internet of Things



Kevin Ashton



Internet of Things

[Our Brands](#) ▾

[Our Impact](#) ▾



[Our Story](#) ▾

[Coupons ↗](#)

The Tide logo is prominently displayed on the left side of the banner. In the center, there is a portrait photo of Kevin Ashton. To the right of the photo, his name is written in blue, followed by a large, bold, white text block containing the words "A FORCE FOR GOOD", "A FORCE FOR GROWTH", and "2018 CITIZENSHIP REPORT".

Tide

Kevin Ashton **A FORCE FOR GOOD
A FORCE FOR GROWTH
2018 CITIZENSHIP REPORT ↗**



Internet of Things

[Our Brands](#) ▾

[Our Impact](#) ▾



[Our Story](#) ▾

[Coupons ↗](#)

The background features the Tide logo on the left and a portrait of Kevin Ashton in the center. The P&G website navigation bar is visible at the top.

Tide

Kevin Ashton A FORCE FOR GOOD
A FORCE FOR GROWTH
2018 CITIZENSHIP REPORT ↗



1972

1982

1999

Internet of Things

[Our Brands](#) ▾

[Our Impact](#) ▾

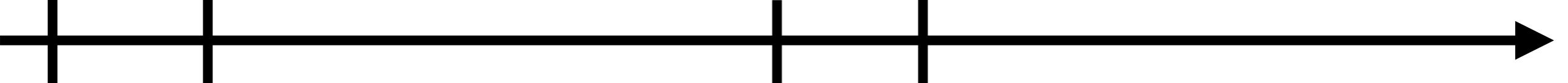


[Our Story](#) ▾

[Coupons ↗](#)

Tide

Kevin Ashton **A FORCE FOR GOOD**
A FORCE FOR GROWTH
2018 CITIZENSHIP REPORT ↗



1972 1982

1999

2003

Arduino

- Programmable device.
- Easily connectable.
- Open source.
- Simple to use software.

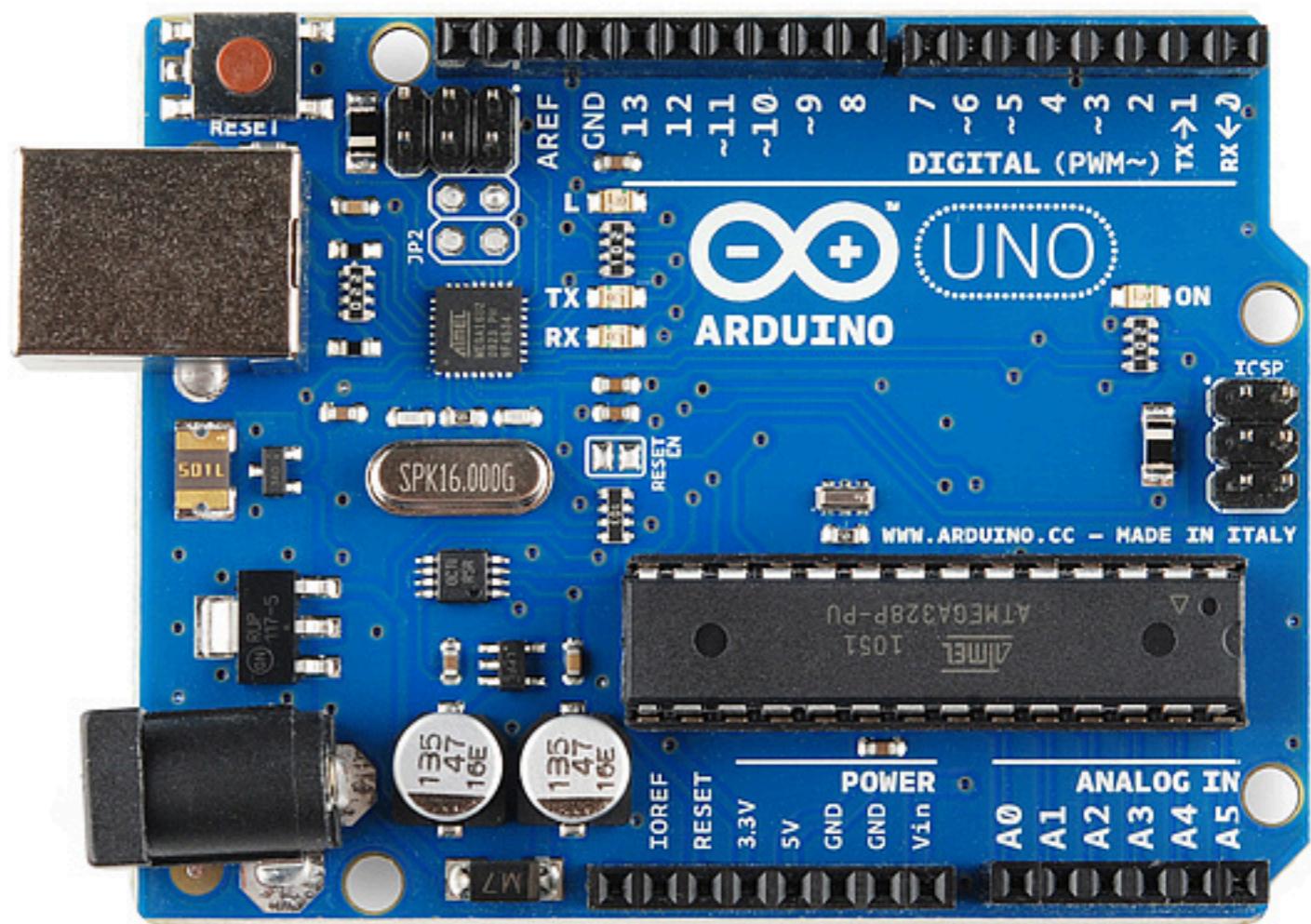
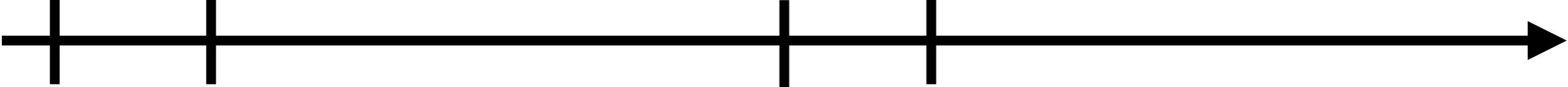


Image source: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>



1972 1982

1999

2003

Arduino

- Programmable device.
- Easily connectable.
- Open source.
- Simple to use software.

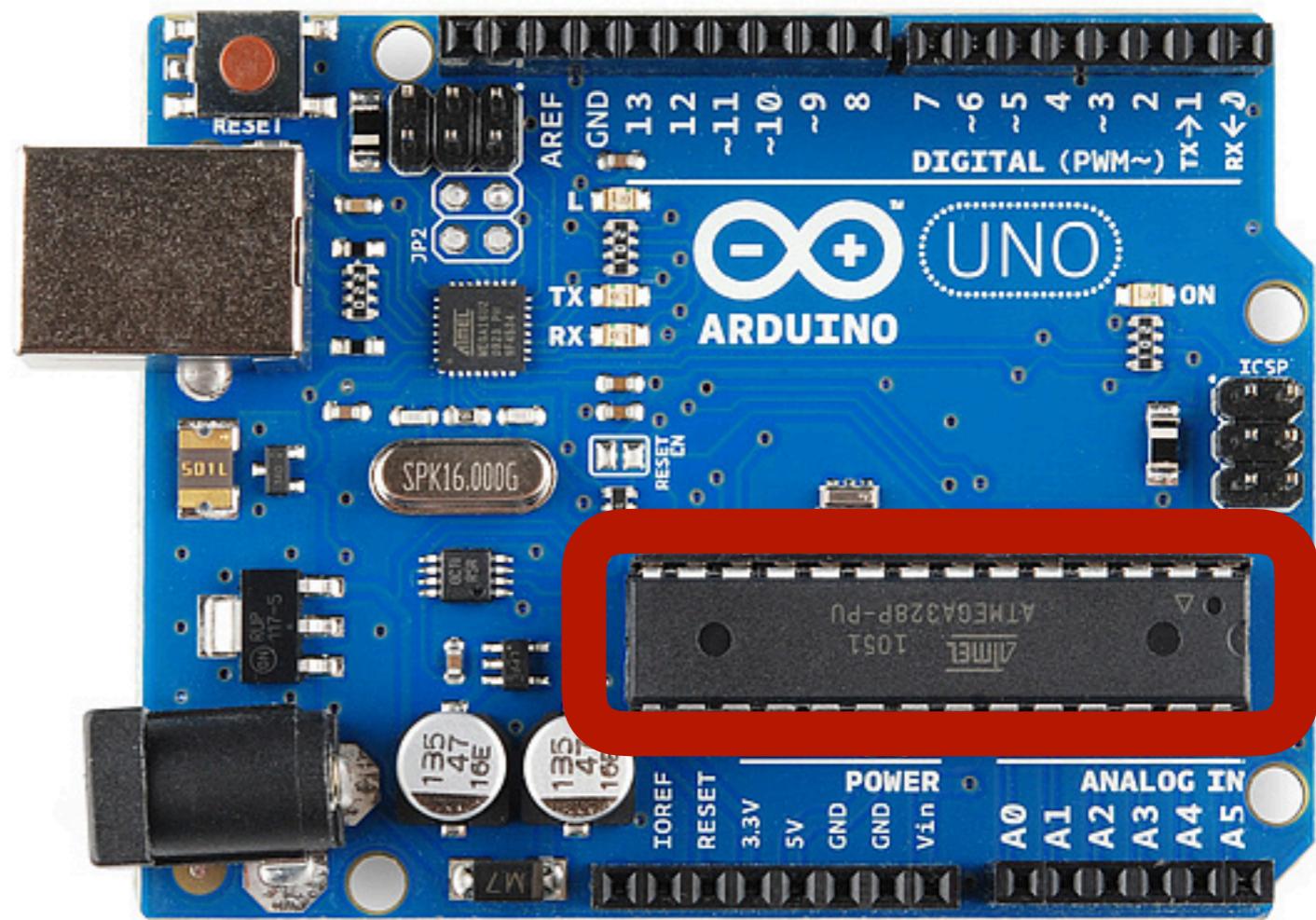
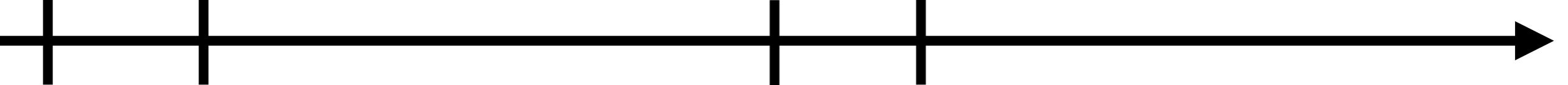


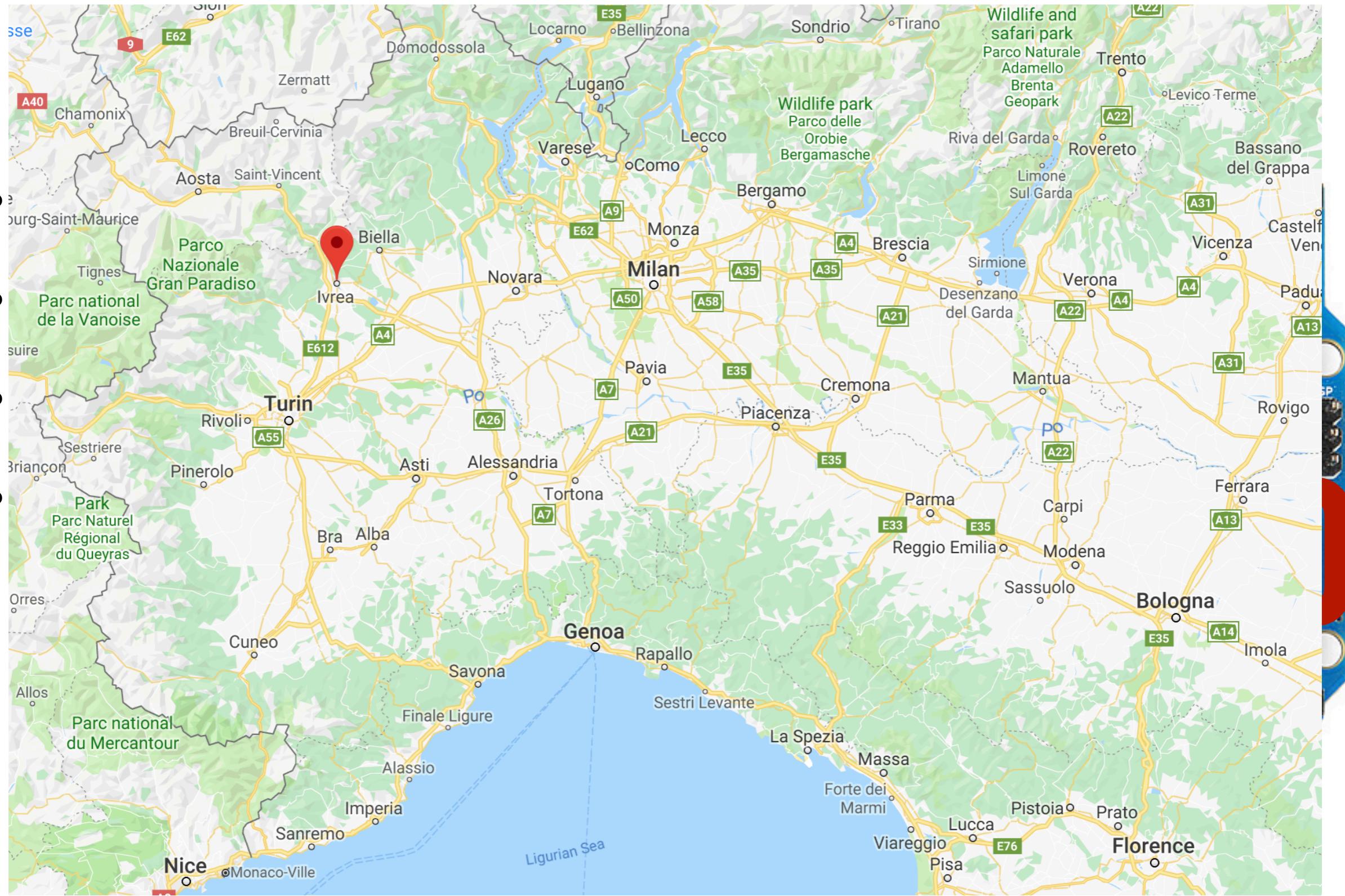
Image source: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>

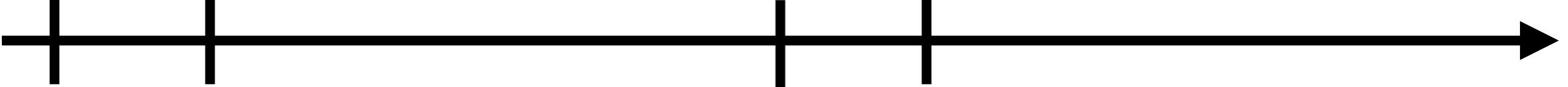


1972 1982

1999 2003

Arduino





1972 1982

1999

2003

Arduino

The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0.3". The code editor displays the "Blink" sketch. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

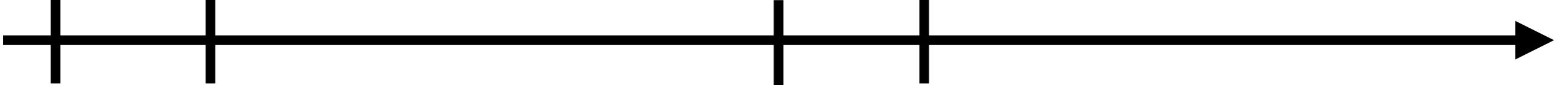
  This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop(){
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

At the bottom of the screen, there is a black status bar with the text "Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz".



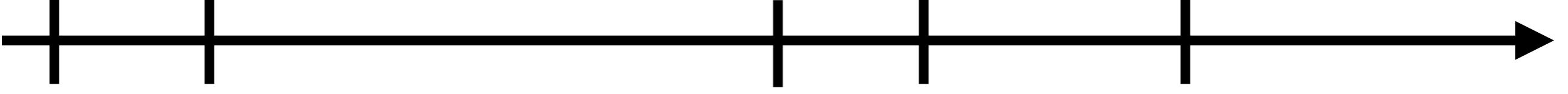
1972 1982

1999 2003

Arduino Ethernet

- Programmable device.
- Easily connectable.
- Open source.
- Simple to use software.
- Has network connectivity.





1972 1982 1999 2003 2012

Raspberry Pi

- Computer.
- Linux inside.
- Programming oriented.
- Full network stack.

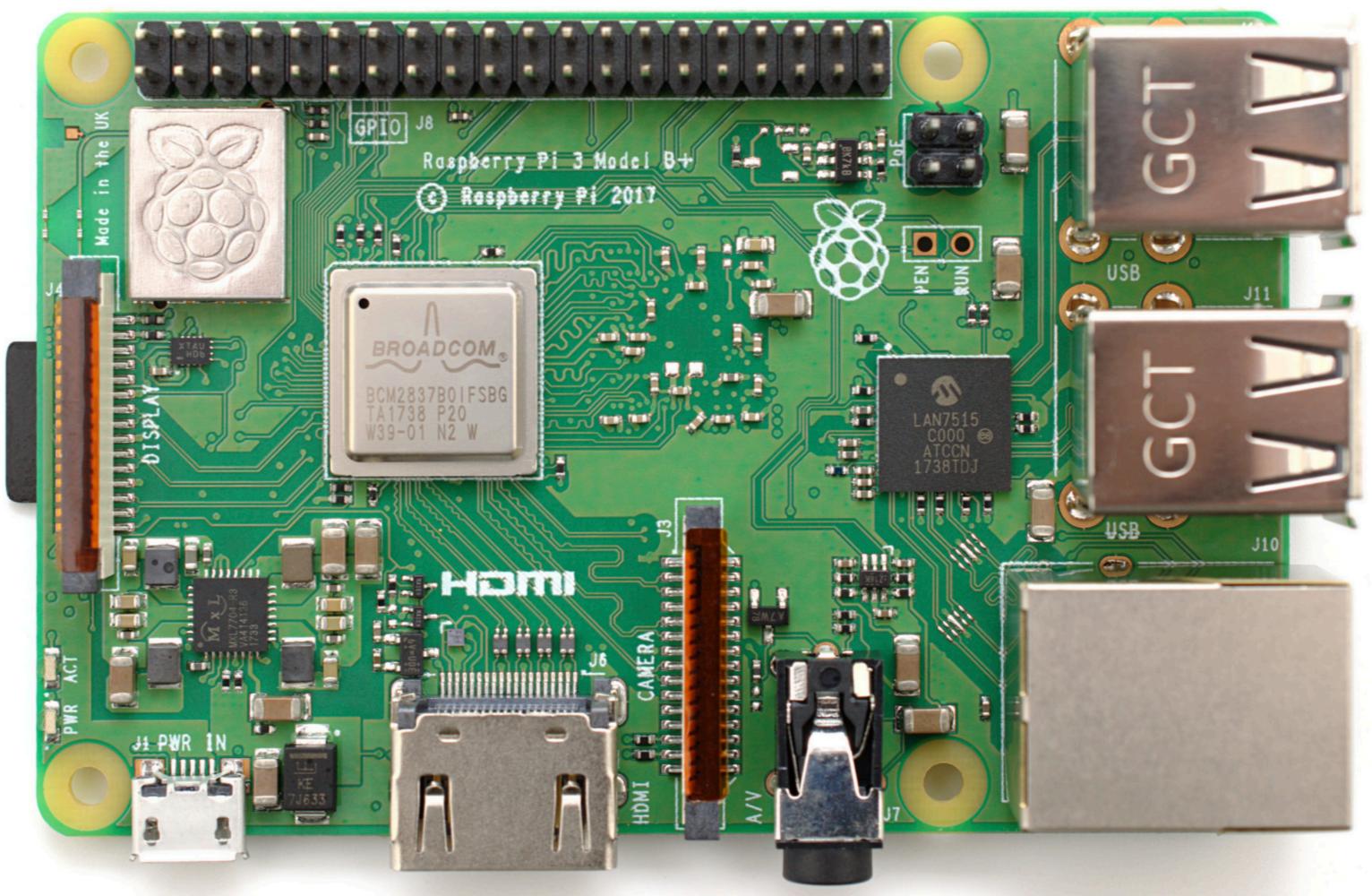
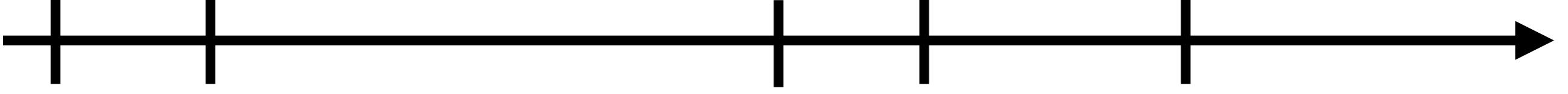


Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



1972 1982 1999 2003 2012

Raspberry Pi

- Computer.
- Linux inside.
- Programming oriented.
- Full network stack.

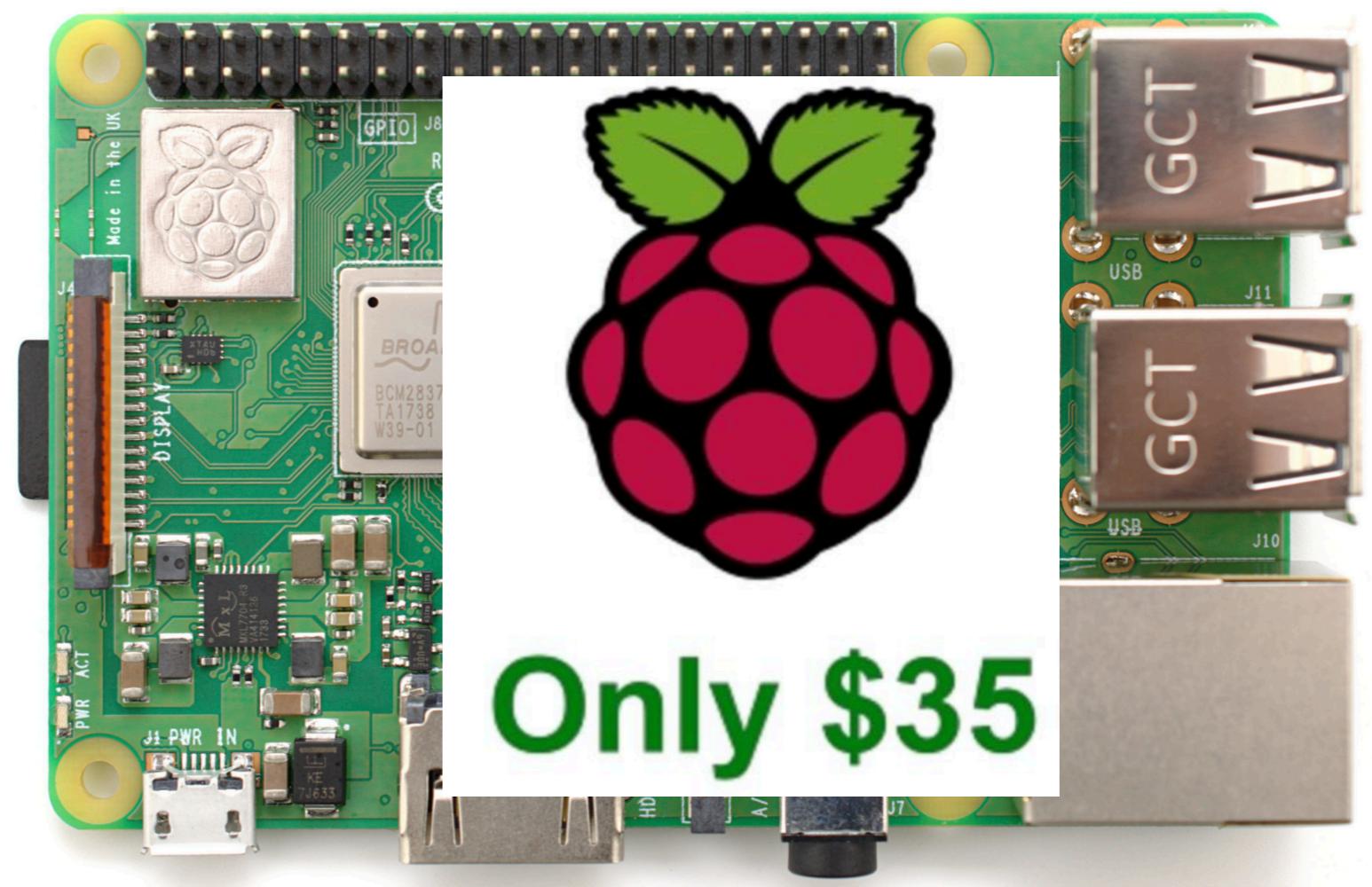
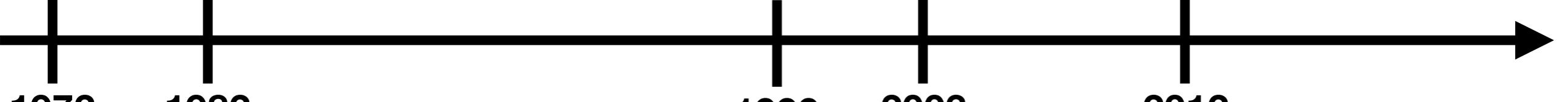


Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



Raspberry Pi

February 29th, 2012

- Computer.
- Linux inside.
- Programming oriented.
- Full network stack.

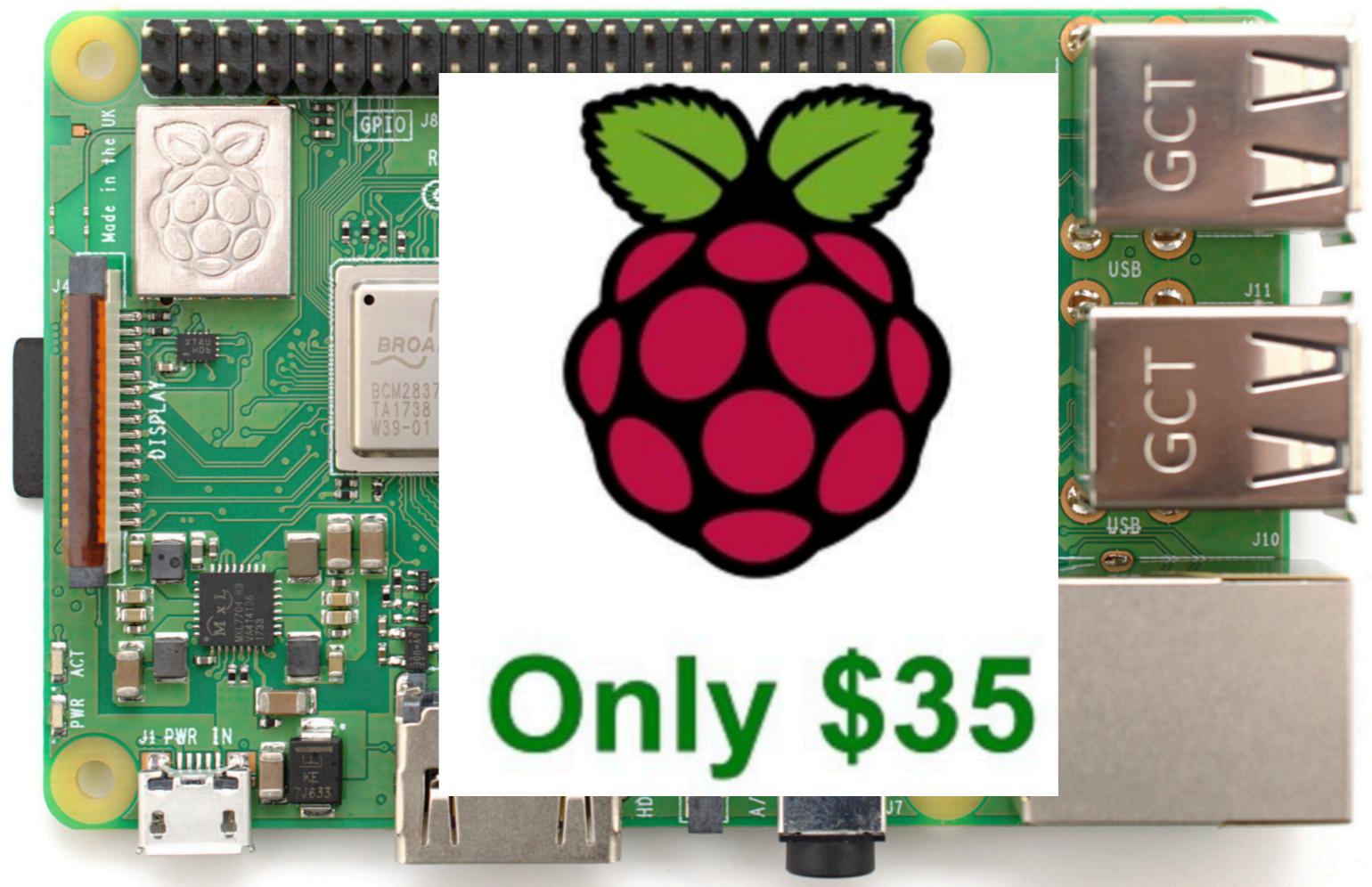
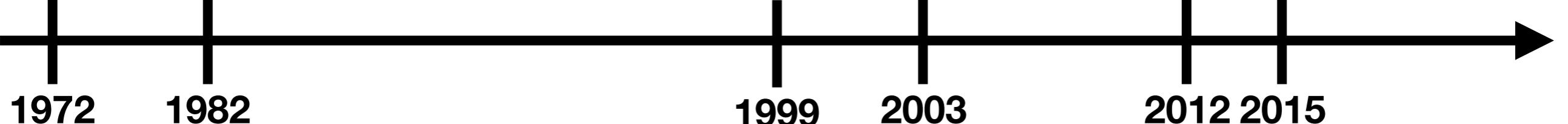


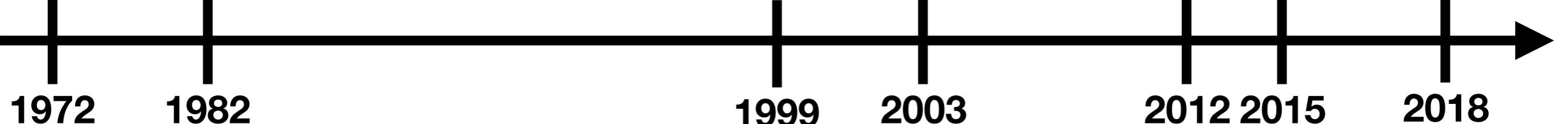
Image source: <https://www.flickr.com/photos/120586634@N05/39906369025/>



Raspberry Pi Zero

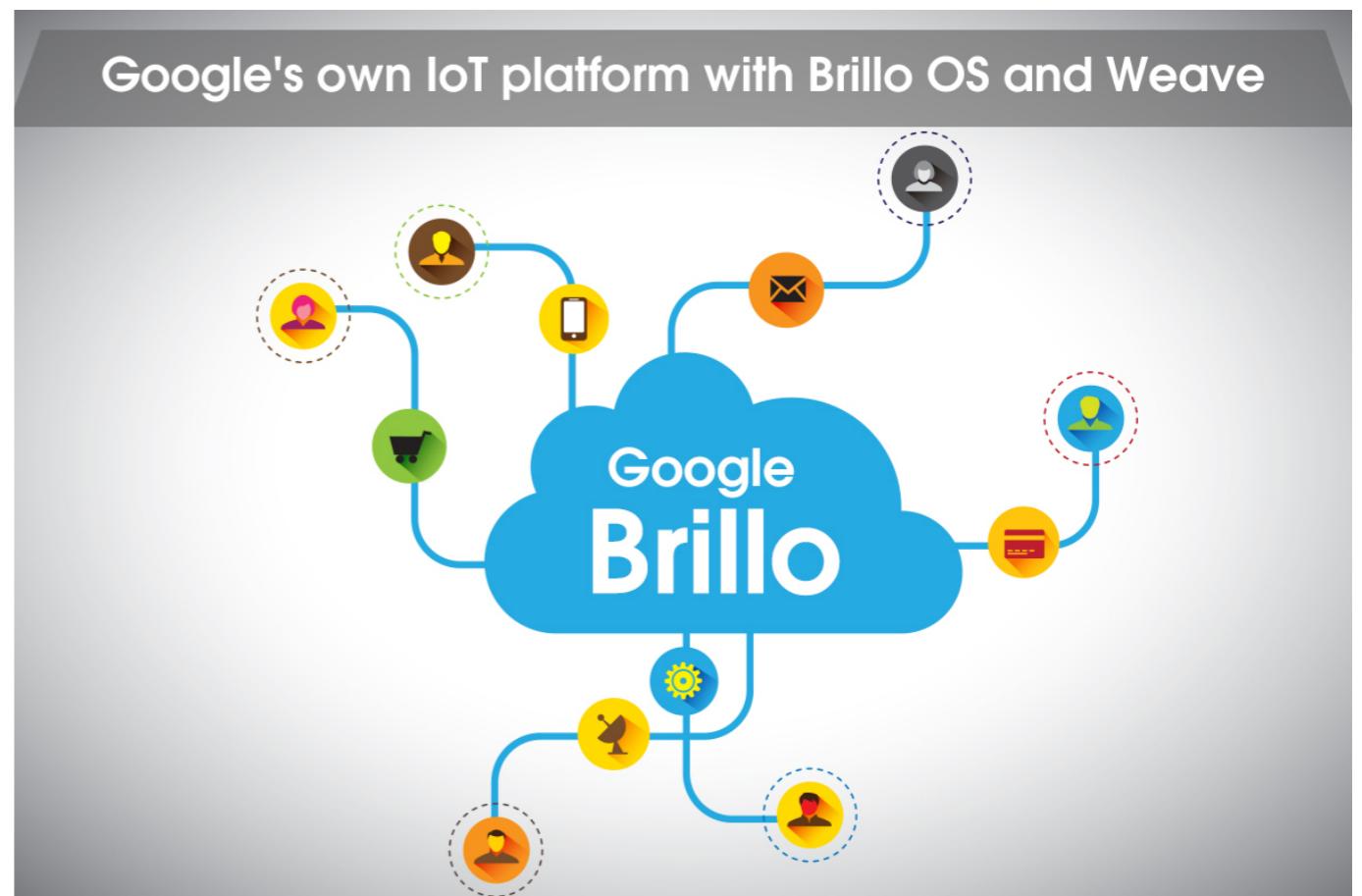
- A Broadcom BCM2835 application processor
- 1GHz ARM11 core
 - 40% faster than Raspberry Pi 1
- 512MB of LPDDR2 SDRAM
- A micro-SD card slot
- A mini-HDMI socket for 1080p60 video output
- Micro-USB sockets for data and power
- An unpopulated 40-pin GPIO header
- Small form factor, at 65mm x 30mm x 5mm





Android Things

- Android Things lets you experiment with building devices on a trusted platform, without previous knowledge of embedded system design:
 - Develop using the Android SDK and Android Studio.
 - Access hardware such as displays and cameras natively through the Android framework.
 - Connect your apps with Google services.
 - Integrate additional peripherals through the Peripheral I/O APIs.
- Use the Android Things Console to push over-the-air feature and security updates.



Good with Sensors

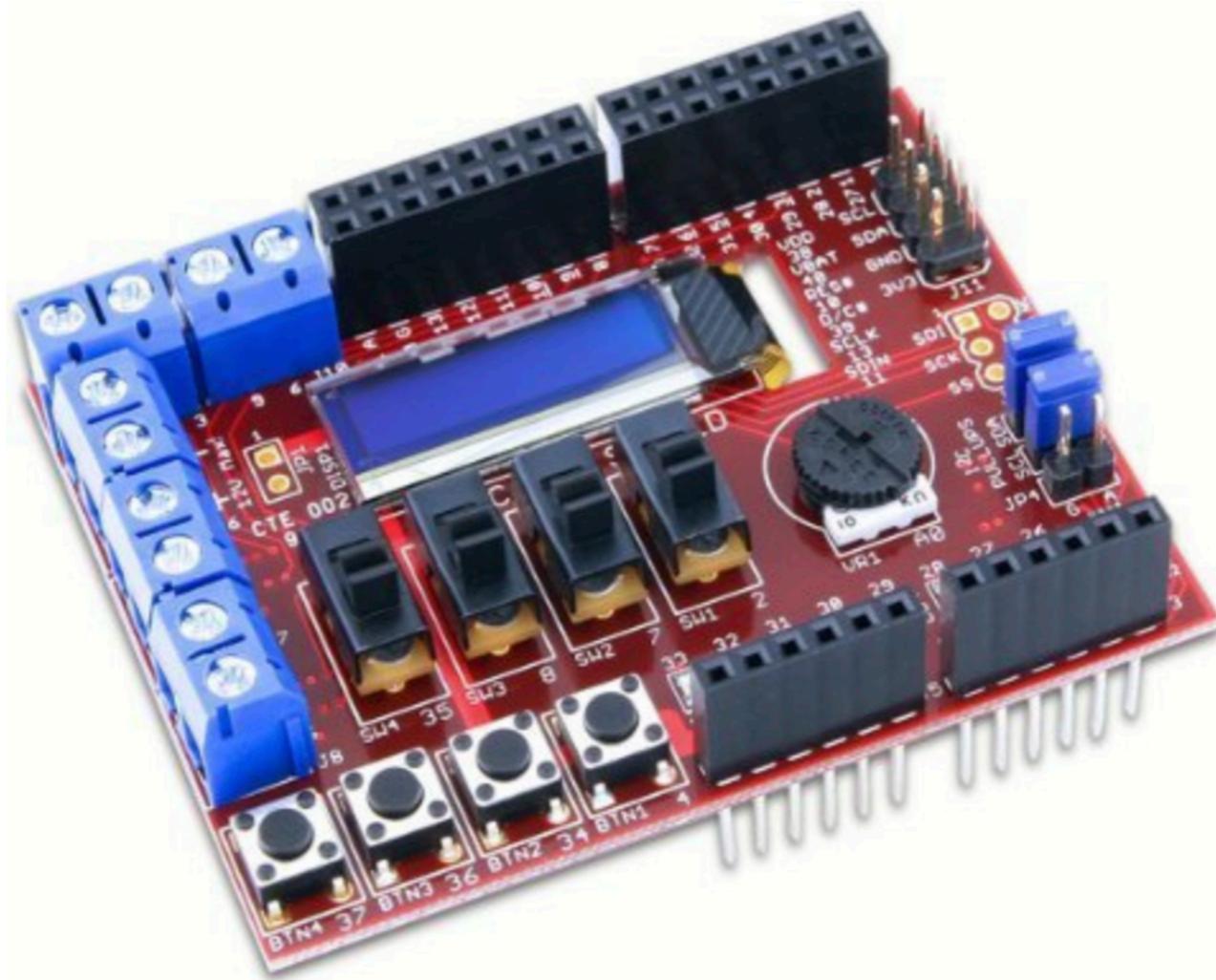


Arduino Uno

20\$

ATmega328P

Good with Sensors

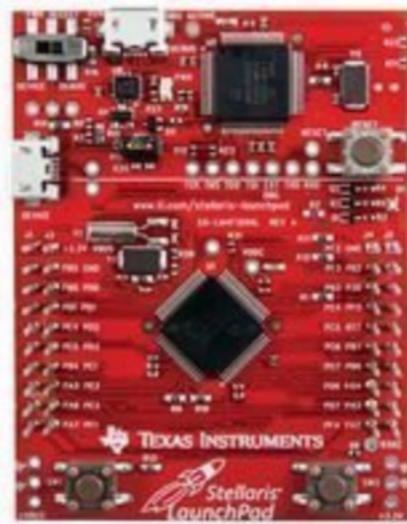


chipKIT Basic I/O Shield

40\$

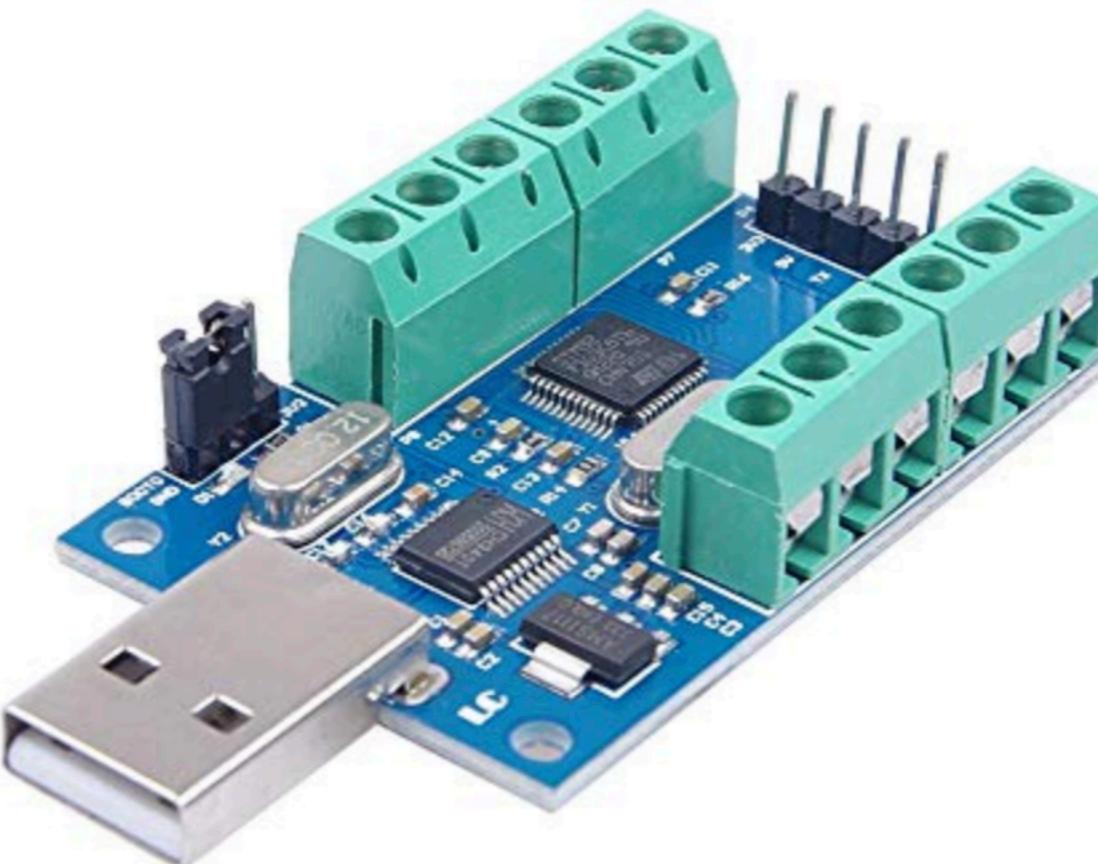
PIC

Good with Sensors



LAUNCHPAD
25\$
LM4F120

Good with Sensors and for Processing

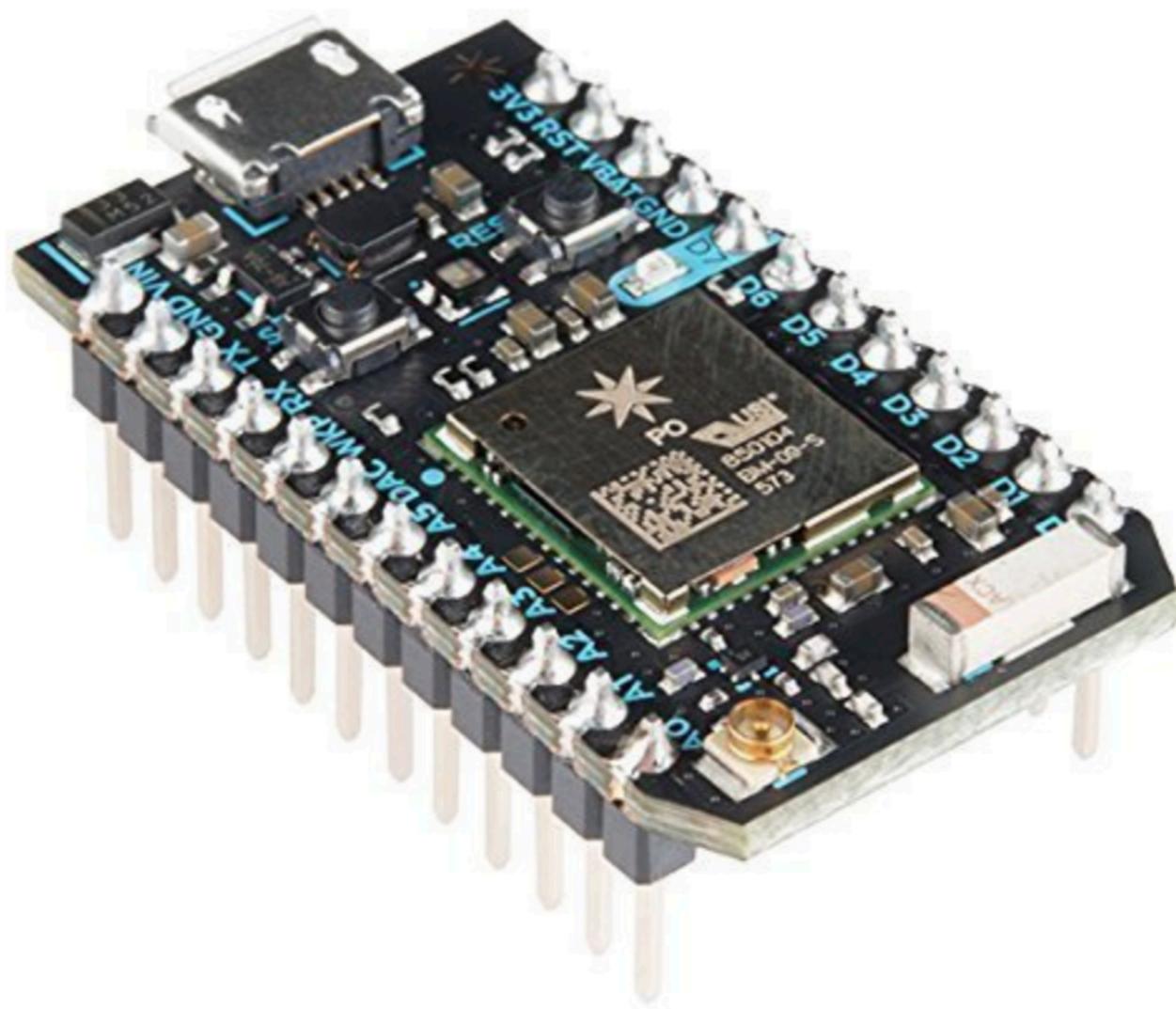


STM32

13\$

CH340 ADC Module

Good with Sensors and for Processing



Particle

20\$

ARM

Good with Sensors and for Processing

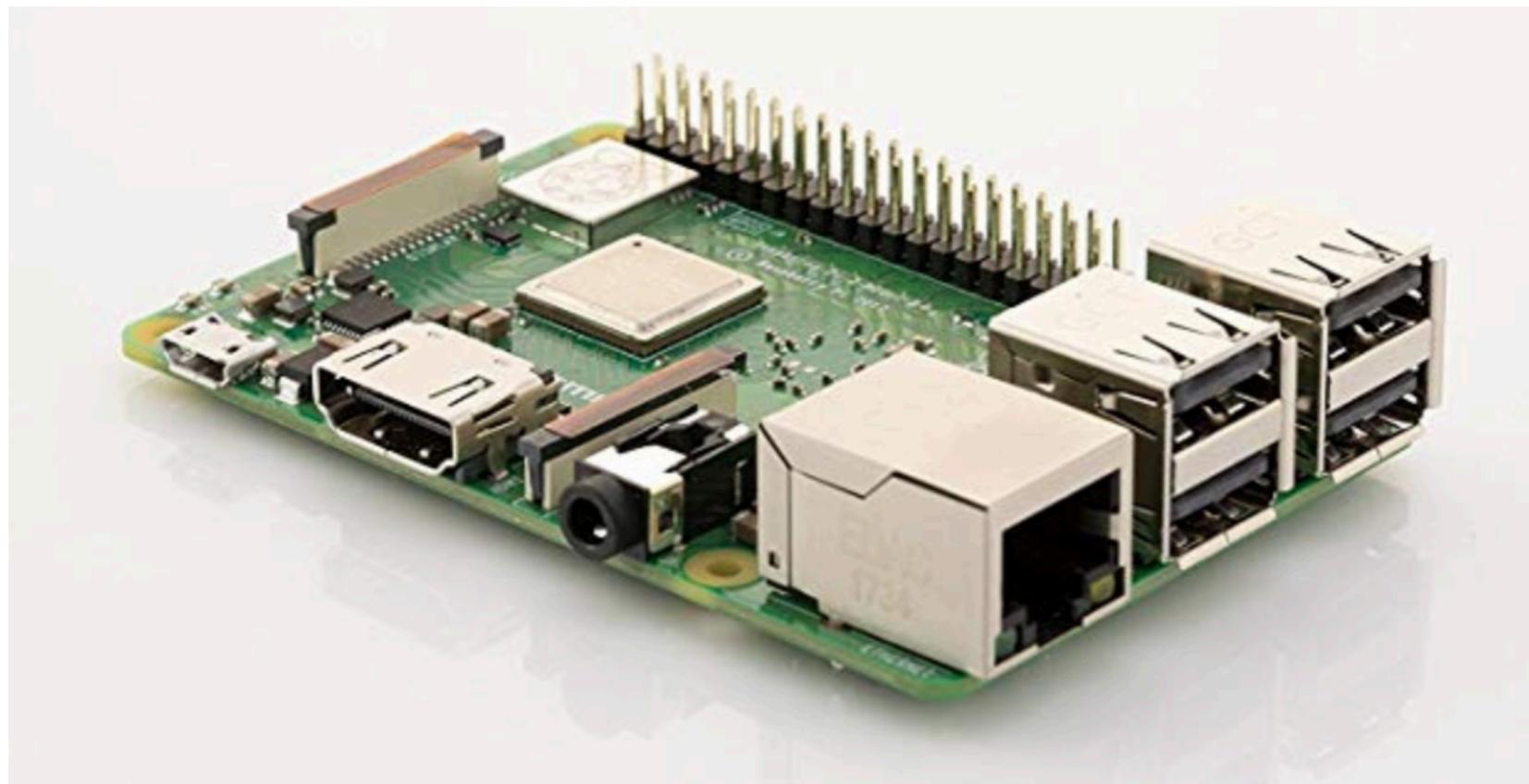


Raspberry Pi Zero

5\$

ARM

Good for Processing and Network

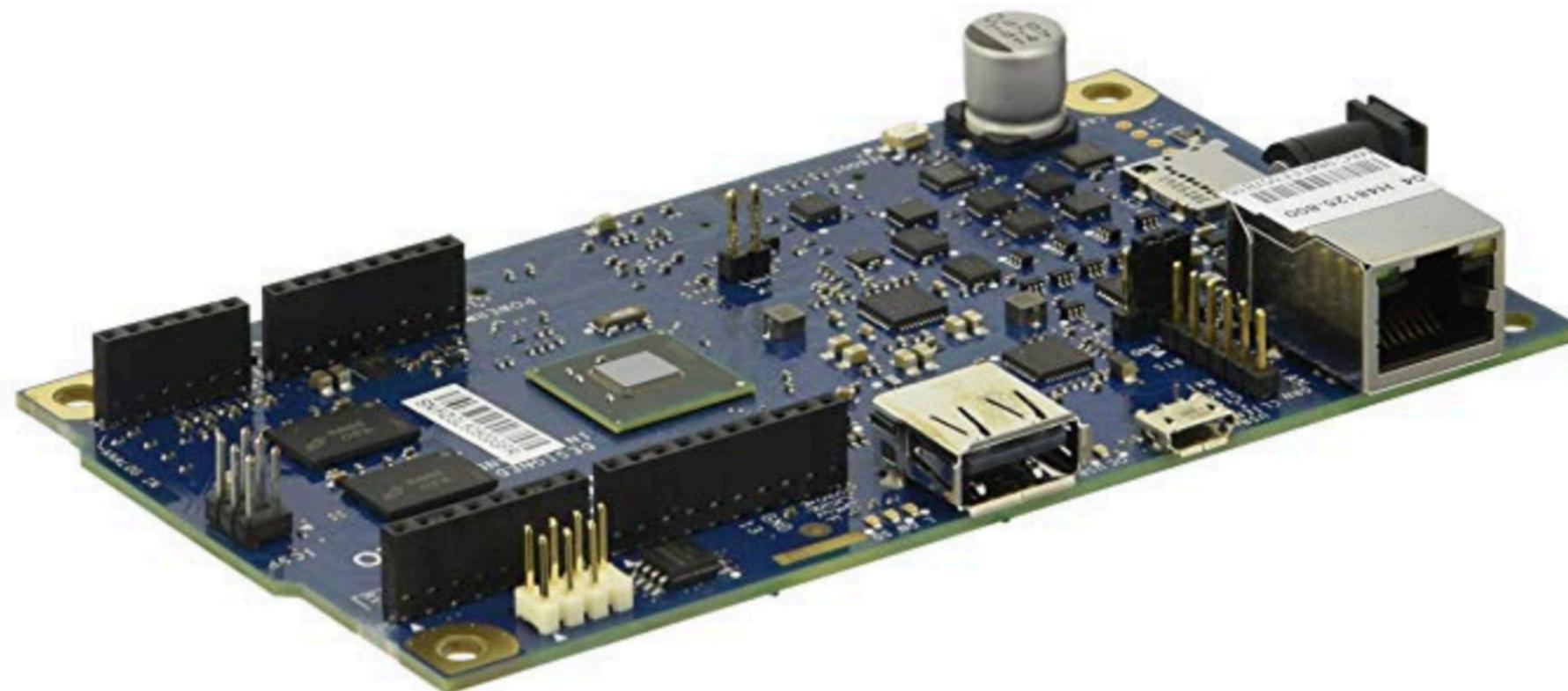


Raspberry Pi 3 B+

35\$

ARM - 1GB RAM

Good for Processing and Network



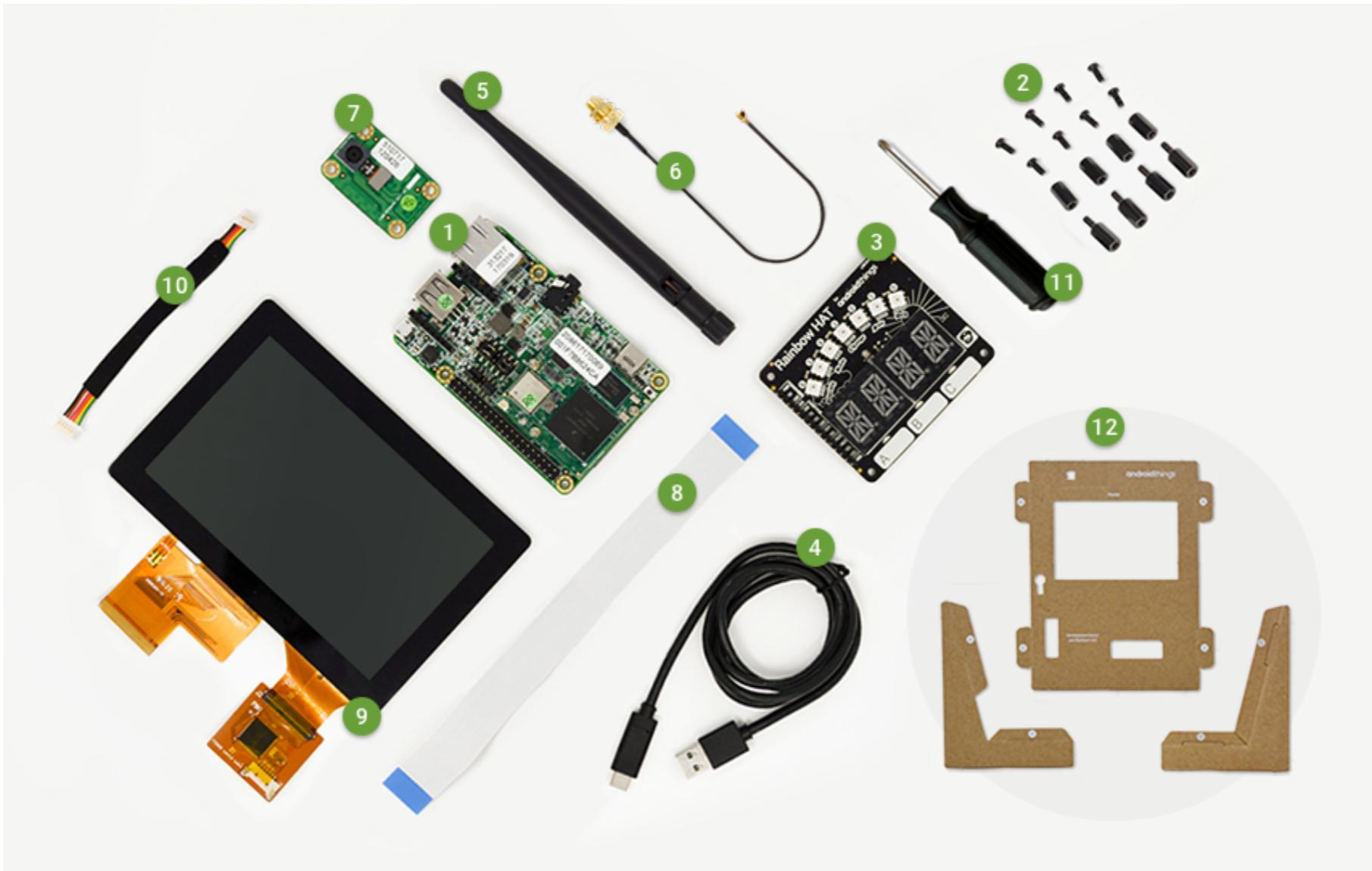
Intel Galileo Gen 2P

50\$

Quark SoC - 400MHz - 256MB RAM

Android Things

NXP i.MX7D Starter Kit



ARM Cortex-A7 + M4

200\$

<https://shop.technexion.com/pico-pi-imx7-startkit-rainbow-hat.html>

Android Things Raspberry Pi Kit



ARM Cortex A53

100\$

<https://androidthings.withgoogle.com/#!kits/raspberry-pi-3-starter-kit>

IDE Options

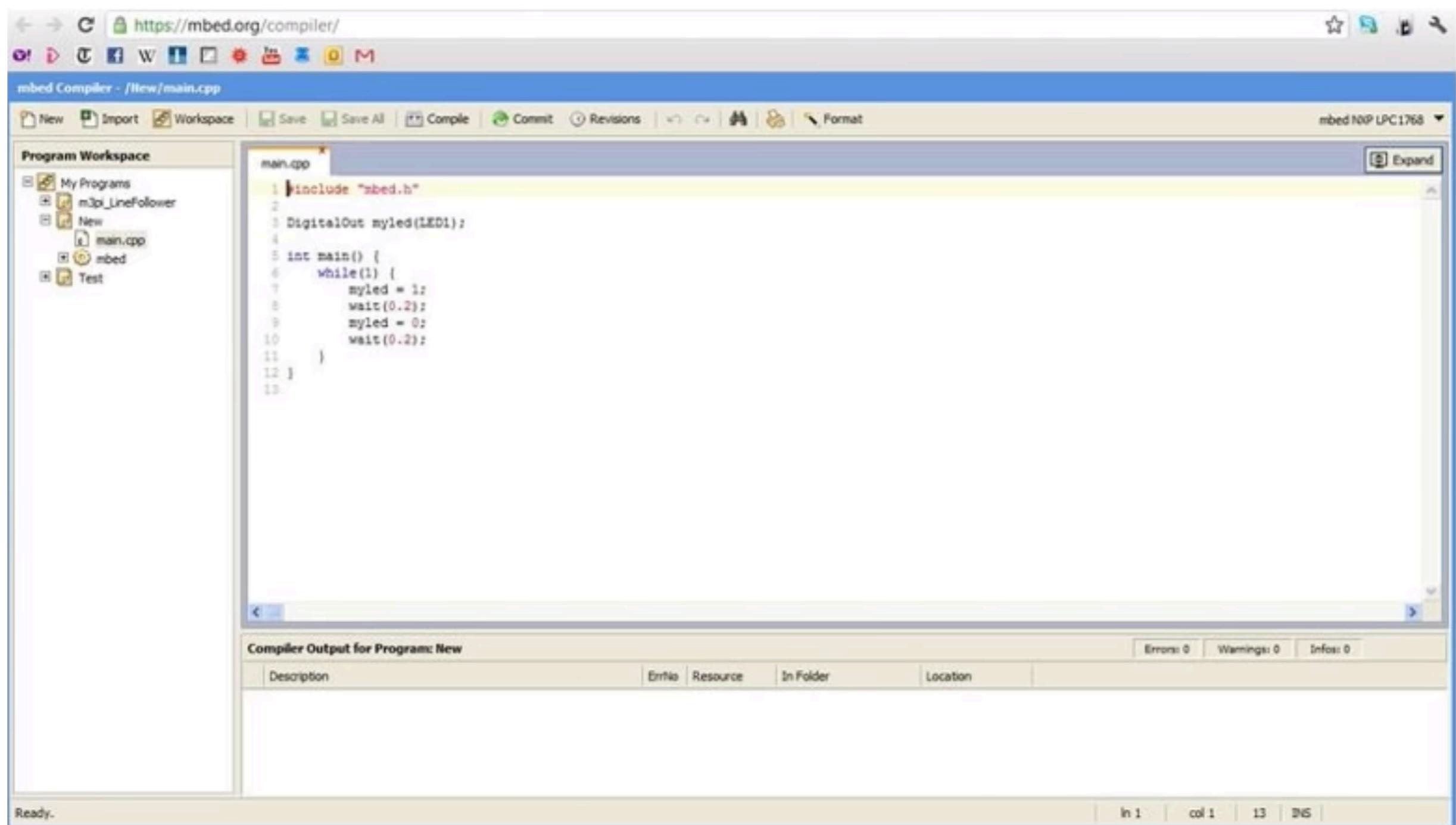
The screenshot shows the Arduino IDE's Sketchbook interface. On the left, a teal sidebar menu includes options like EDITOR, Sketchbook (which is selected), Examples, Libraries, Monitor, Help, and Preferences. The main area displays a sketch titled "sketch_feb24a". At the top right, there are buttons for NEW SKETCH, SEARCH SKETCHBOOK, and various sharing options. Below the title, it says "sketch_feb24a.ino" and "ReadMe.adoc". The code editor shows the following sketch:

```
/*
 */
void setup() {
}
void loop() {
}
```

Below the code editor, a large upload icon is present, along with the text: "Import your sketches to your online Sketchbook and access them from any device!".

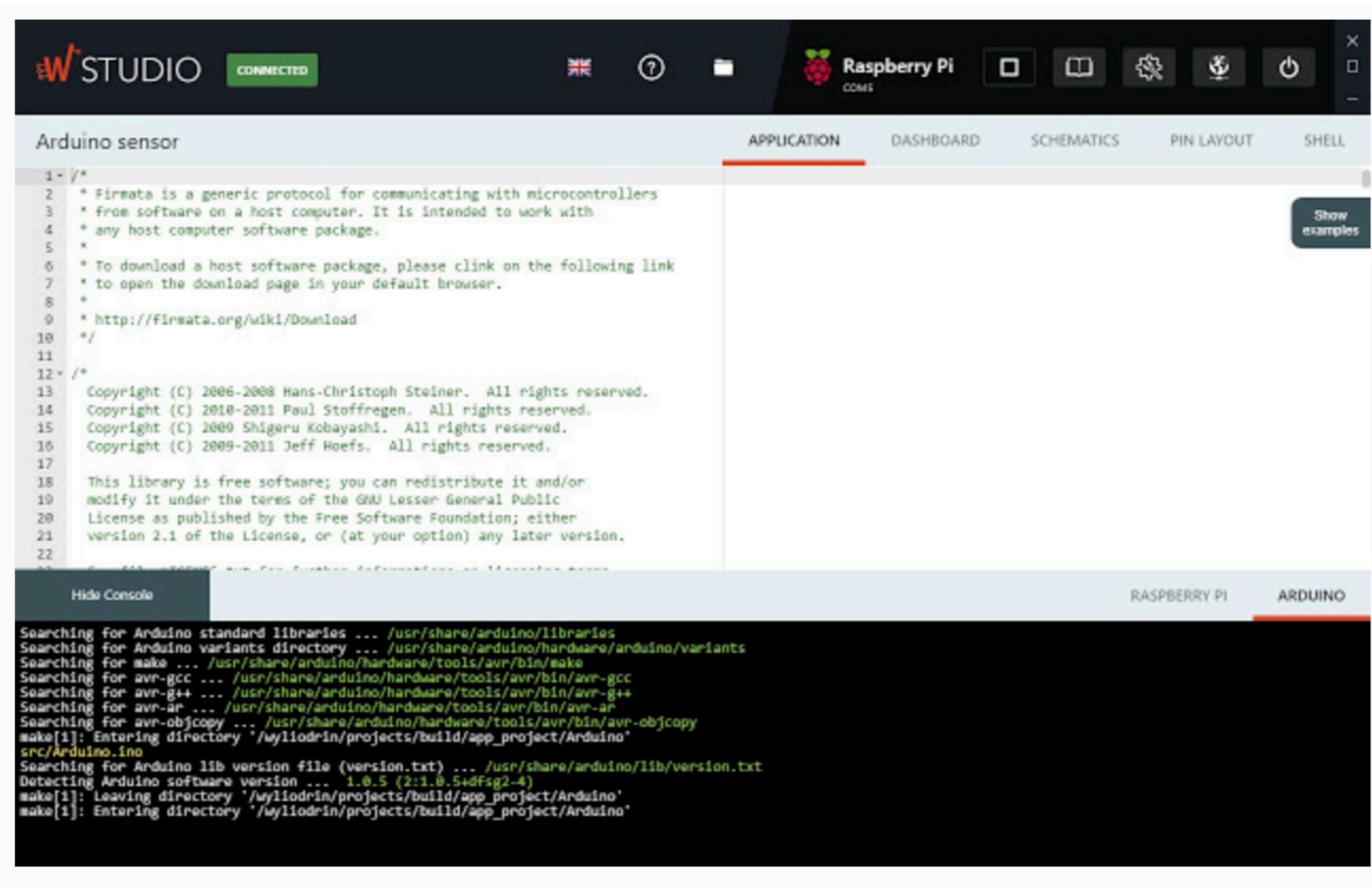
<https://create.arduino.cc>

IDE Options



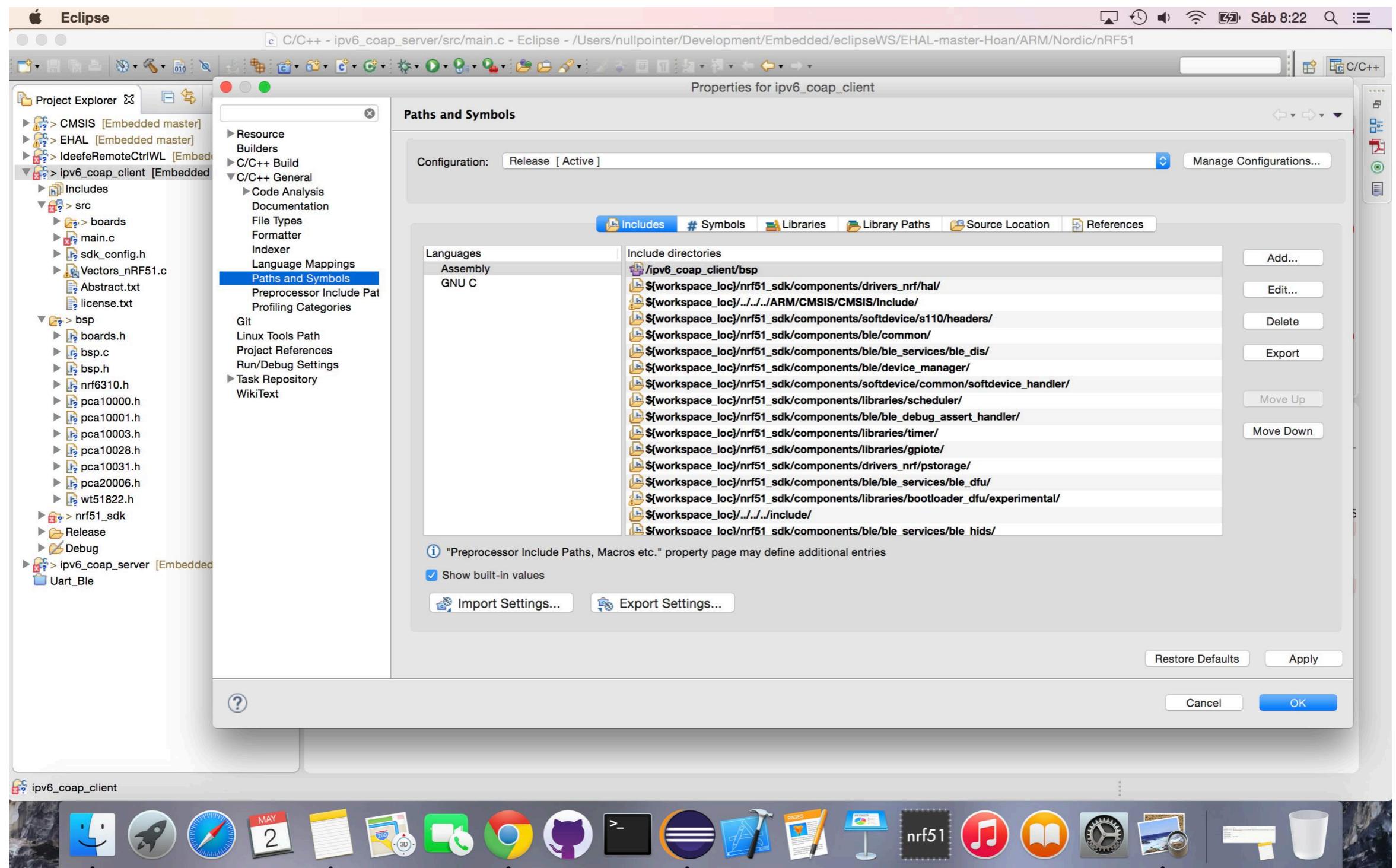
<https://os.mbed.com/>

IDE Options

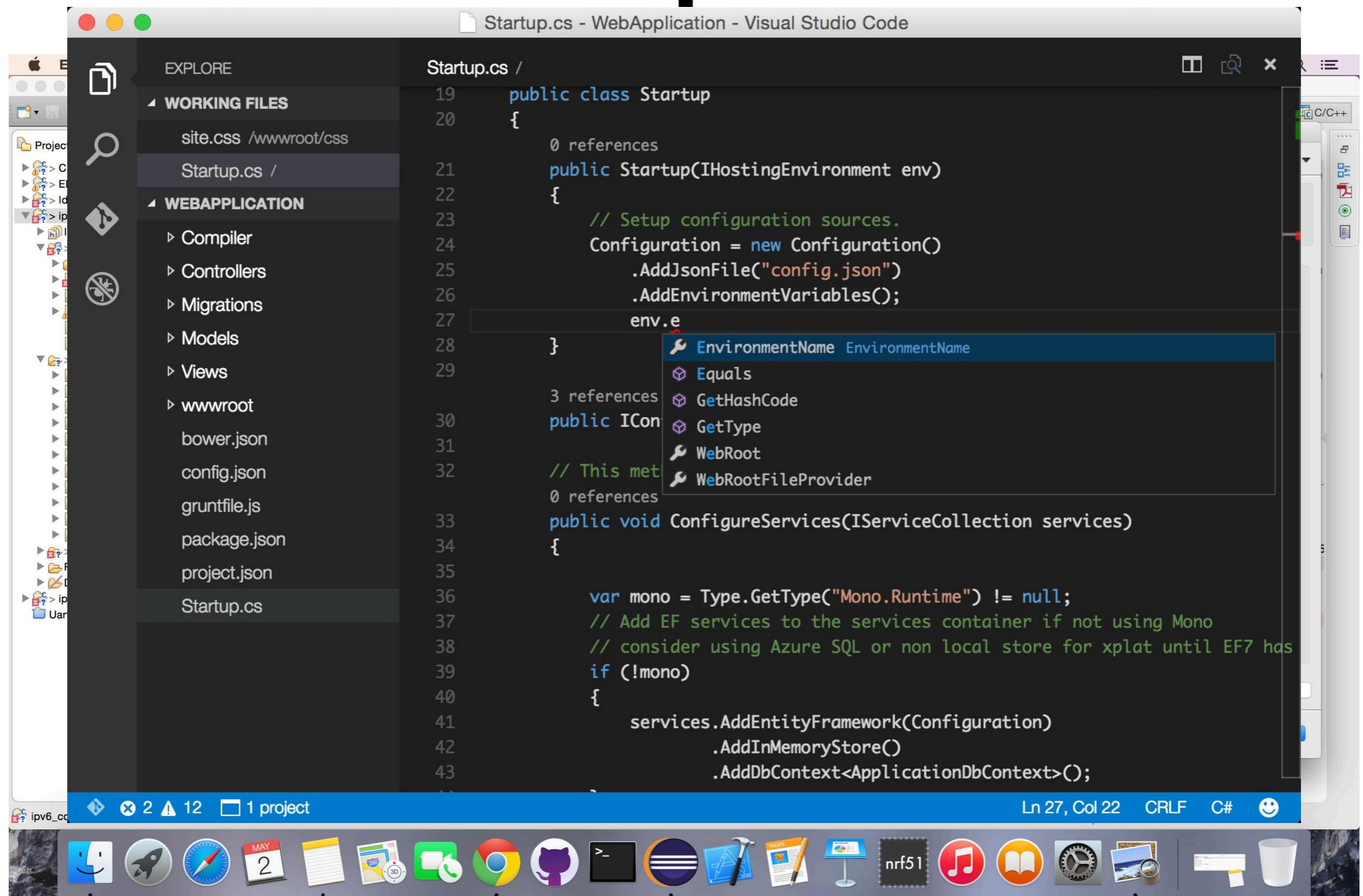


<https://wyliodrin.com/>

IDE Options



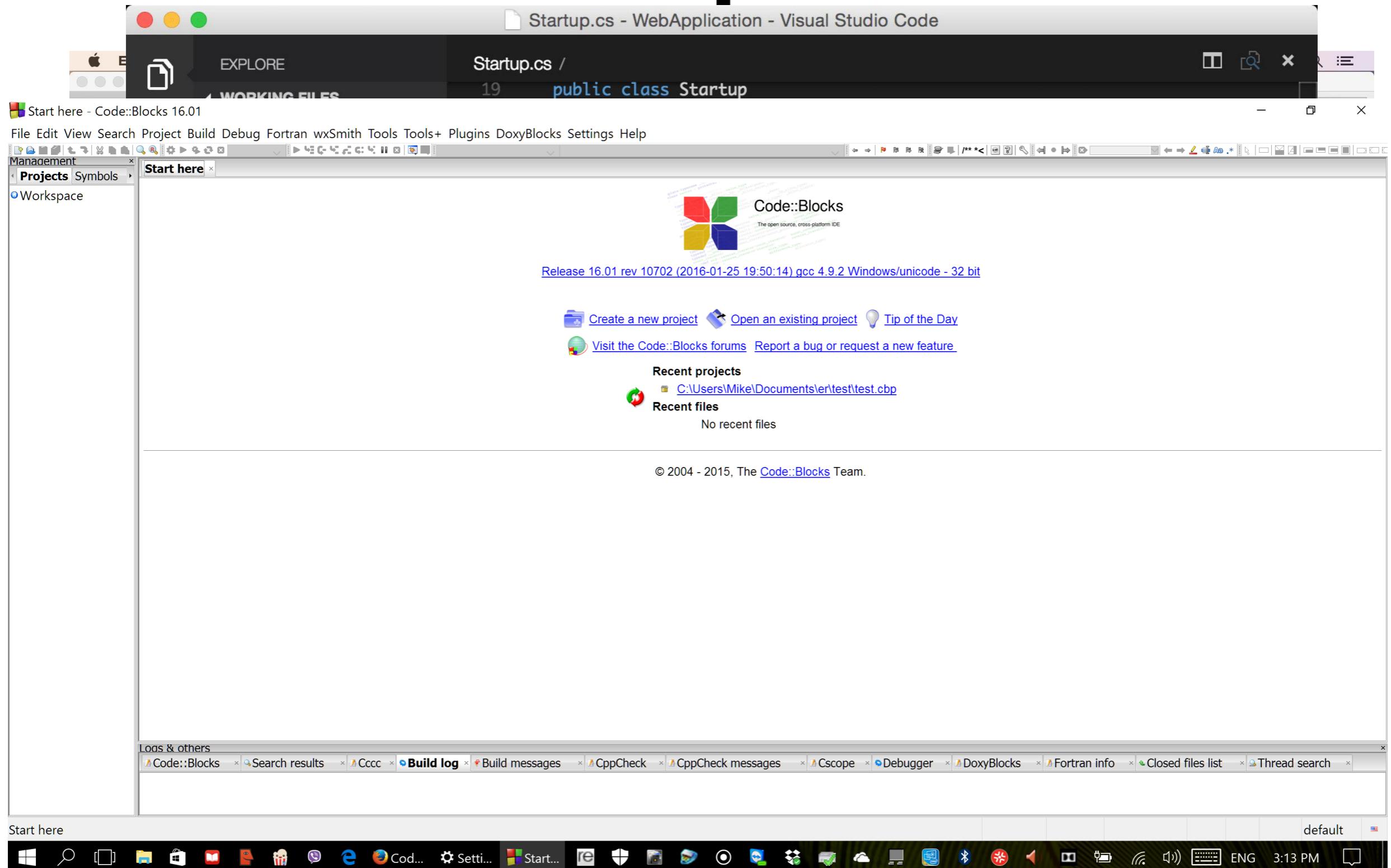
IDE Options



A screenshot of the Visual Studio Code interface. The title bar says "Startup.cs - WebApplication - Visual Studio Code". The left sidebar shows a project structure with files like site.css, Startup.cs, config.json, etc. The main editor window displays C# code for a Startup class. A code completion dropdown is open at line 27, showing suggestions for 'env.e'. The suggestions include 'EnvironmentName', 'Equals', 'GetHashCode', 'GetType', 'WebRoot', and 'WebRootFileProvider'. The status bar at the bottom shows "Ln 27, Col 22" and "CRLF". The bottom tray has various icons for file operations and system status.

```
public class Startup
{
    public Startup(IHostingEnvironment env)
    {
        // Setup configuration sources.
        Configuration = new Configuration()
            .AddJsonFile("config.json")
            .AddEnvironmentVariables();
        env.e
    }
    public IConfiguration Configuration { get; }
    // This method gets called by the runtime. Use this method to add services to the container.
    public void ConfigureServices(IServiceCollection services)
    {
        var mono = Type.GetType("Mono.Runtime") != null;
        // Add EF services to the services container if not using Mono
        // consider using Azure SQL or non local store for xplat until EF7 has
        if (!mono)
        {
            services.AddEntityFramework(Configuration)
                .AddInMemoryStore()
                .AddDbContext<ApplicationContext>();
        }
    }
}
```

IDE Options

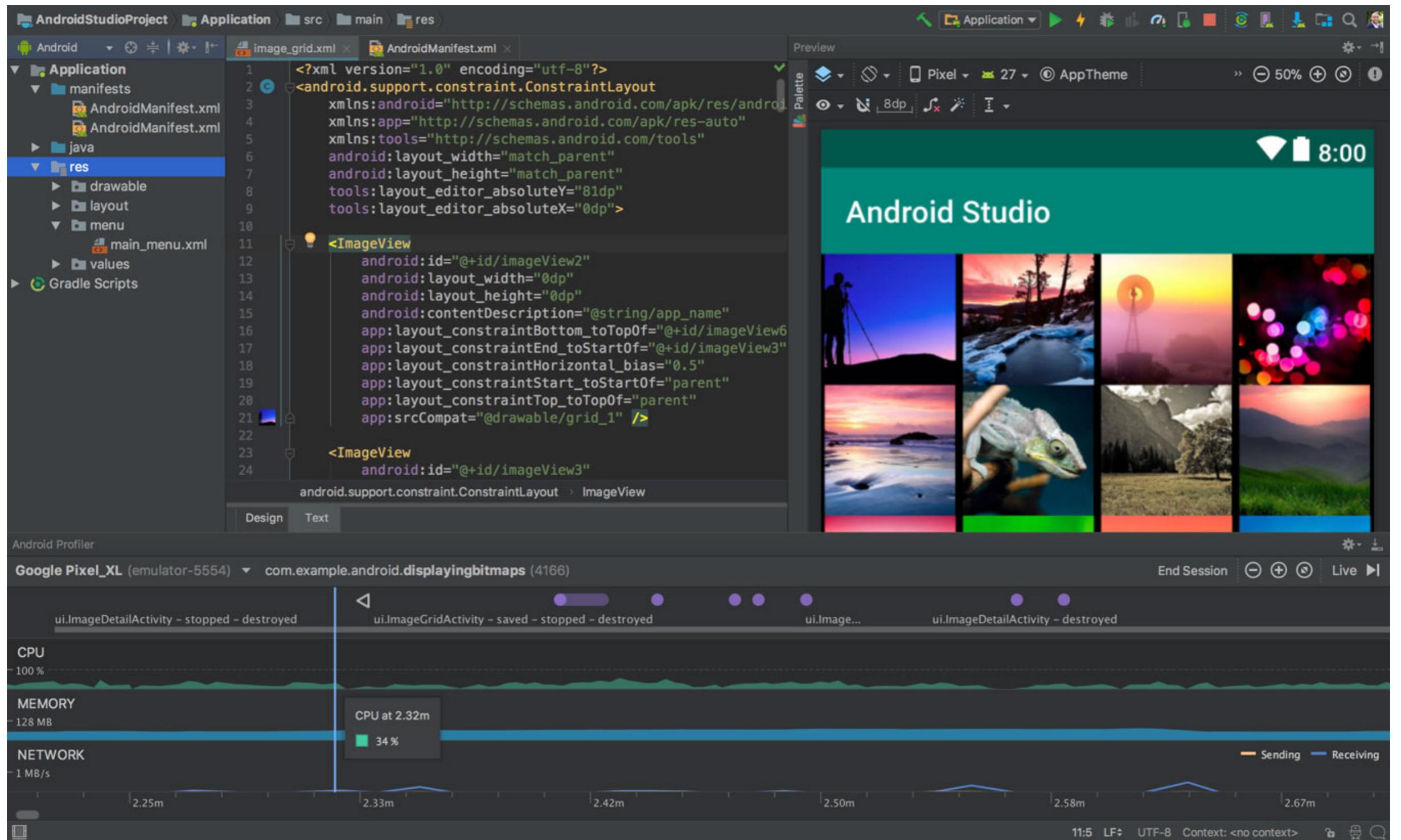


IDE Options

The screenshot shows a terminal window with the following details:

- File Tree:** On the left, a file tree displays the directory structure of a Ruby application. It includes paths like controllers/, helpers/, jobs/, mailers/, models/, views/, static_pages/, bin/, config/, db/, lib/, log/, public/, test/, and controllers/. A file named _footer.html.erb is currently selected.
- Code Editor:** The main area shows a Ruby test file named `Startup.cs`. The code is a test suite for a controller, specifically `StaticPagesControllerTest`. The `setup` method is highlighted in yellow. The code uses RSpec and Capybara assertions.
- Status Bar:** At the bottom, the status bar shows the current mode as `NORMAL`, the file name as `<_controller_test.rb[+]`, and the command `< setup < ruby << 100% : 36: 1`. It also indicates the file is part of `static_pages_controller_test.rb`.
- Bottom Line:** The bottom line shows the current window titles: [4] 0:ruby 1:ruby 2:vim* 3:fish- 4:irssi# 5:mutt and the date/time: [Thu Aug 18] 02:15 pm |

IDE Options



<https://developer.android.com/studio>

Laboratory Projects

- Two projects:
 - Individual project - 60% of the final grade.
 - Team project - 40% of the final grade.



Image source:
<http://inkawall.com>

Individual Project



- A project similar to the samples available here:
 - <https://androidthings.withgoogle.com/build/#!/samples/>
 - Or enhance an existing sample.

Image source:
<http://ceelo.org/individual-leader/>

Individual Project

- A project similar to the samples available here:
 - <https://androidthings.withgoogle.com/build/#!/samples/>
- Or enhance an existing sample.
- Due: Last laboratory before Easter holiday.
 - Either **April 18th** or **April 25th**, depending on your laboratory frequency.
- Deliverables:
 - The source code should be hosted in a github classroom repository.
 - A webpage presenting the project results, similar to:
 - <https://androidthings.withgoogle.com/#!/samples/doorbell>
 - A short video presenting the results.



Image source:
<http://ceelo.org/individual-leader/>

Team Project



Image source:
<https://www.contractingbusiness.com>

- A team of 4 or 5 students to tackle a real-world problem.
- Either:
 - Choose an existing project proposed by the lab instructor.
 - Define a new one, together with the lab instructor.

Team Project

- A team of 4 or 5 students to tackle a real-world problem.
- Either:
 - Choose an existing project proposed by the lab instructor.
 - Define a new one, together with the lab instructor.
- Due: Last laboratory.
 - Either **May 16th** or **May 23rd**, depending on your laboratory frequency.
- Deliverables:
 - The source code should be hosted in a github classroom team repository.
 - A webpage presenting the project results, similar to:
 - <https://androidthings.withgoogle.com/#!/samples/doorbell>
 - A short video presenting the results.



Image source:
<https://www.contractingbusiness.com>

Lecture outcomes

- Understand the available hardware and software options.
- IDE options.
- Project details.

