Lecture #5
Arduino

Android Things 2020

Emergency Plan

- Doorbell Sample
  - With at least the following sections:
    - Overview - present in short the idea.
    - Schematics Plan - the plan schema built using Fritzing.
    - Pre-requisites - the list of needed components.
    - Setup and Build Plan - the setup plan containing what we have already as done and what we plan to do next.

www.cs.ubbcluj.ro/~dan/at/labPlan.html

Till Last Laboratory

- Individual Project:
  - A short video presenting the results.
  - A complete page (on GitHub) similar to Doorbell Sample with all the sections filled.
- Team Project:
  - All the requirements from the Individual Project.
  - Plus a companion mobile application to manage the IoT application.
- Everything should be added on GitHub repo.

shutterstock.com • 432183739

www.cs.ubbcluj.ro/~dan/at/labPlan.html

No IoT Hardware

- We can use our mobile phones to act as an IoT device to perform at least one of the following:

1.Collect images.

2.Collect sensor data, like temperature, accelerometers, barometer, etc. Any sensor that is available on the mobile device.

3.Collect GPS data.

4.Collect App usage data, etc.

5.Other: you can suggest something else.

No IoT Hardware

- We can use our mobile phones to act as an IoT device to perform at least one of the following:

  1.Collect images.

  2.Collect sensor data, like temperature, accelerometers, barometer, etc. Any sensor that is available on the mobile device.

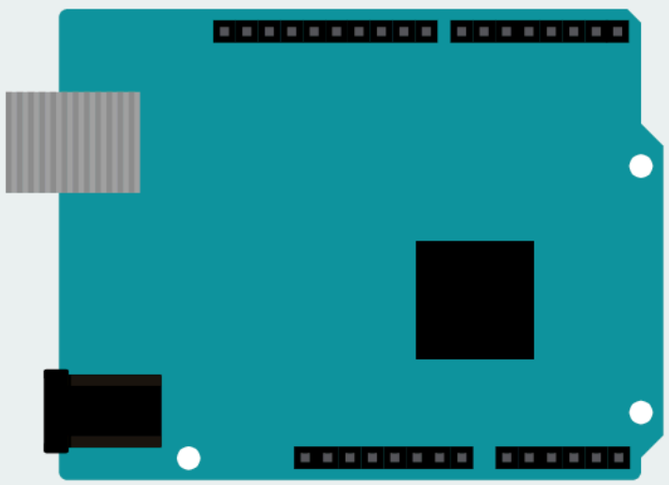
  3.Collect GPS data.

  4.Collect App usage data, etc.

  5.Other: you can suggest something else.

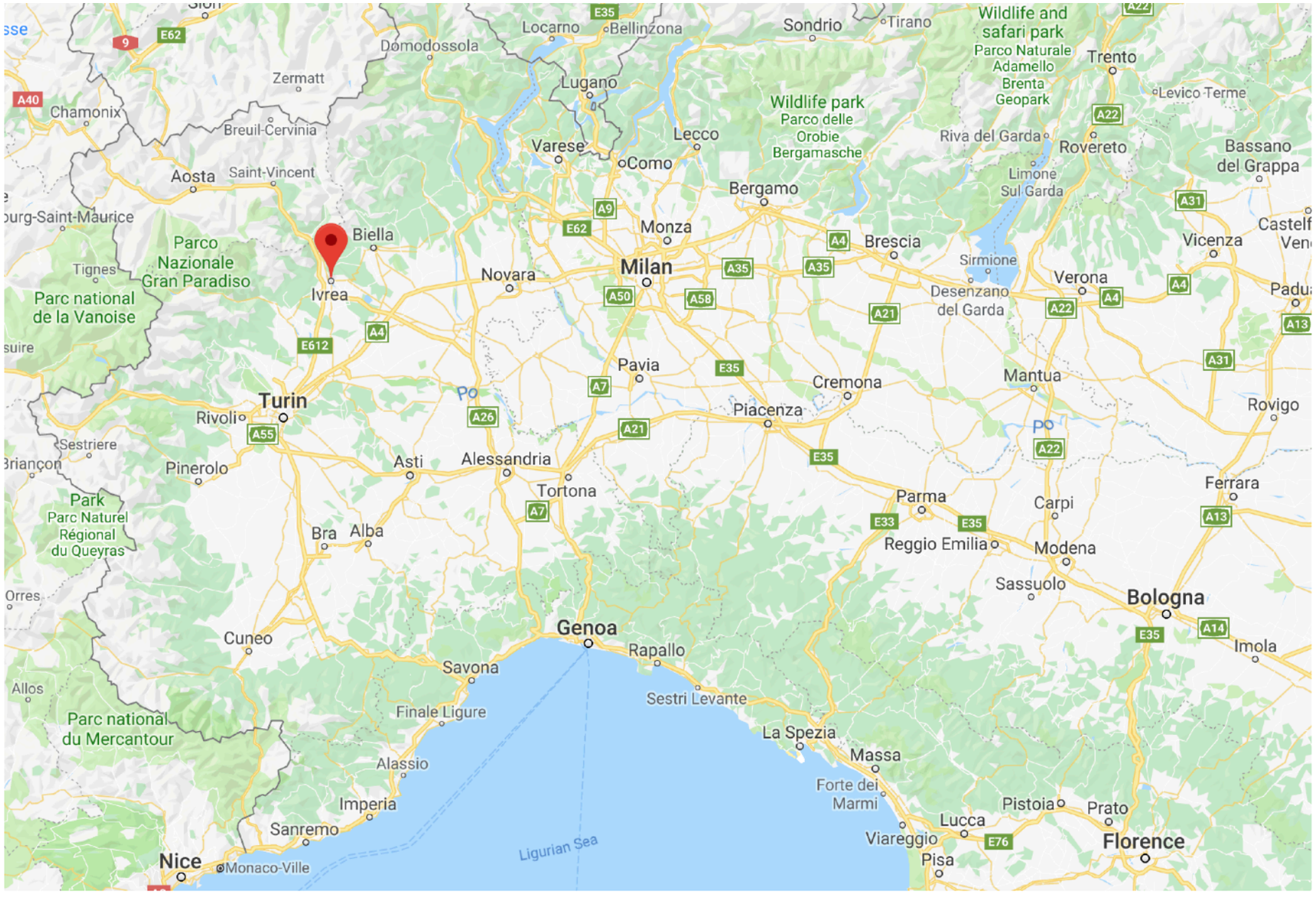
See the ‘No hardware access’ topic on Piazza for more details!

Foundations



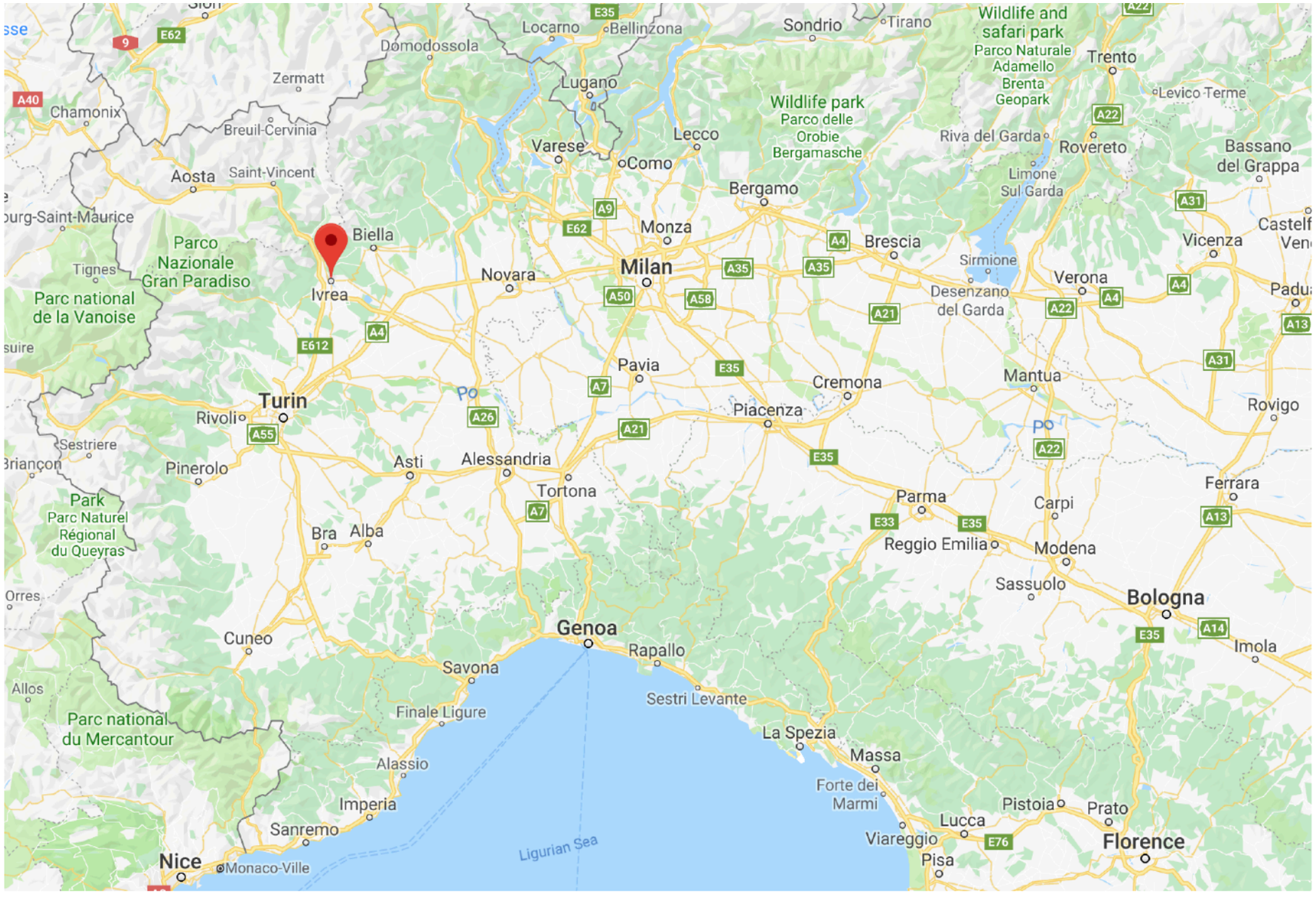
```
void setup() {
}

void loop() {
}
```

History


Locarno
Bellinzona
Sondrio
Tirano
Wildlife and safari park
Parco Naturale Adamello Brenta Geopark
Trento
Levico Terme
Domodossola
Zermatt
Lugano
Chamonix
Breuil-Cervinia
Wildlife park
Parco delle Orobie Bergamasche
Riva del Garda
Rovereto
Varese
Lecco
Como
Aosta
Saint-Vincent
Limone Sul Garda
Bassano del Grappa
Bergamo
Biella
Monza
Brescia
Vicenza
Parco Nazionale Gran Paradiso
Tignes
Milan
Sirmione
Verona
Novara
Ivrea
Parc national de la Vanoise
Desenzano del Garda
Padua
Pavia
Cremona
Mantua
Rovigo
Turin
Rivoli
Piacenza
Po
Sestriere
Pinerolo
Asti
Alessandria
Tortona
Ferrara
Park Parc Naturel Régional du Queyras
Parma
Carpi
Reggio Emilia
Modena
Bra
Alba
Sassuolo
Bologna
Cuneo
Genoa
Rapallo
Imola
Savona
Sestri Levante
Finale Ligure
Parc national du Mercantour
La Spezia
Massa
Alassio
Forte dei Marmi
Pistoia
Prato
Imperia
Lucca
Sanremo
Viareggio
Florence
Ligurian Sea
Nice
Monaco-Ville
Pisa

1972 1982 1999 2003

History



1972 1982 1999 2003

History

Arduino d'Ivrea
(*Civica raccolta stampe - Milano*)

1972 1982 1999 2003

History

Arduino d'Ivrea
(*Civica raccolta stampe - Milano*)

King of Italy (1002–1014)

1972 1982 1999 2003

History

Arduino d'Ivrea
(*Civica raccolta stampe - Milano*)

King of Italy (1002–1014)

https://en.wikipedia.org/wiki/Arduin_of_Ivrea

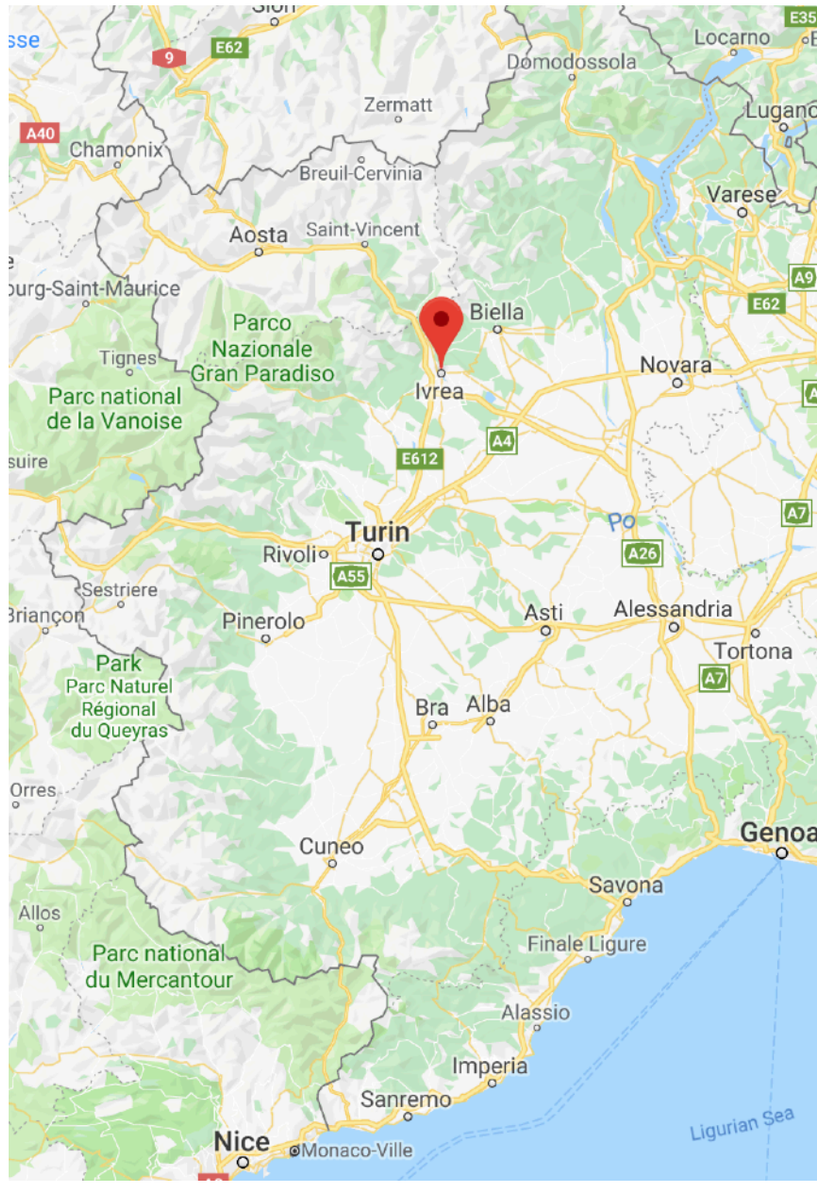
History

Hernando Barragán

History

Hernando Barragán

History



Hernando Barragán

Interaction Design Institute Ivrea
IDII

History

History

Massimo Banzi

History

Casey Reas

Massimo Banzi

History

Casey Reas

Massimo Banzi

2001
https://processing.org/

History




```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

History



```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

History



Wiring

2003

```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

History



Wiring

2003

```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

History



Wiring

2003

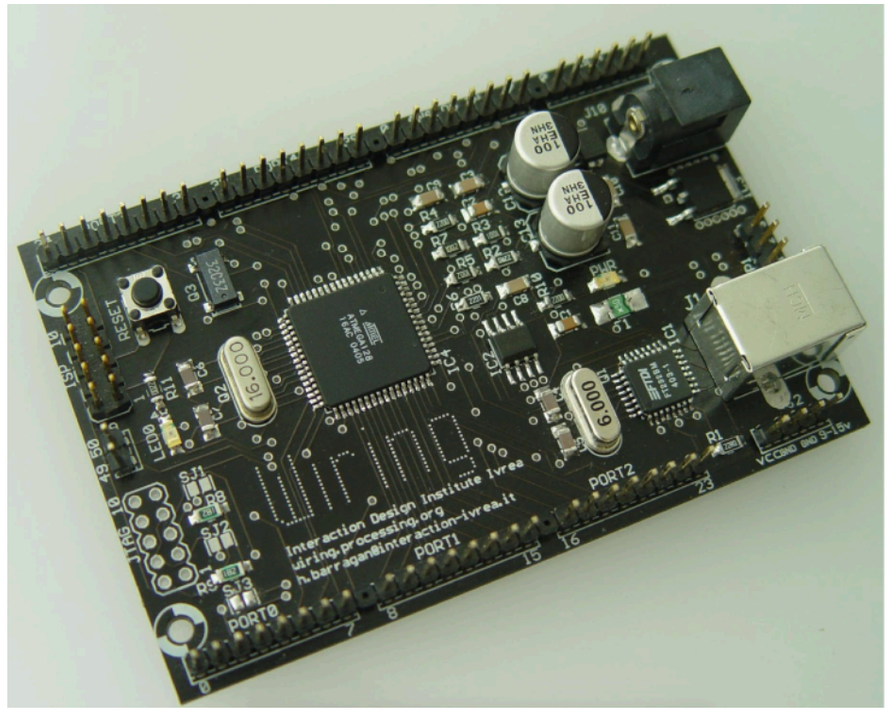
```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

http://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf

History





Wiring

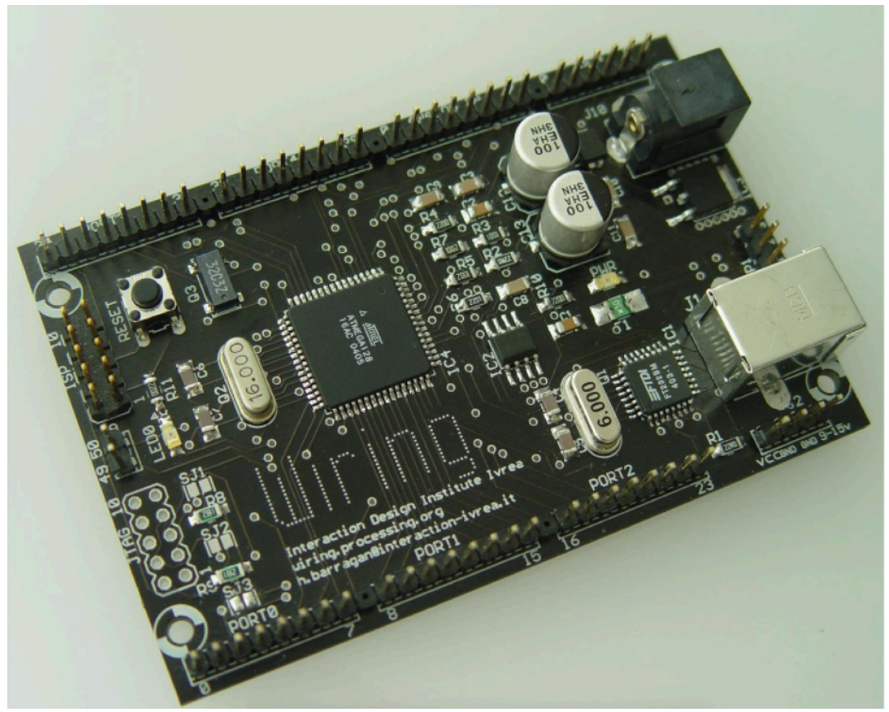
2003

```
void setup() {
  size(400, 400);
  stroke(255);
  background(192, 64, 0);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}
```

```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

http://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf

History



Wiring

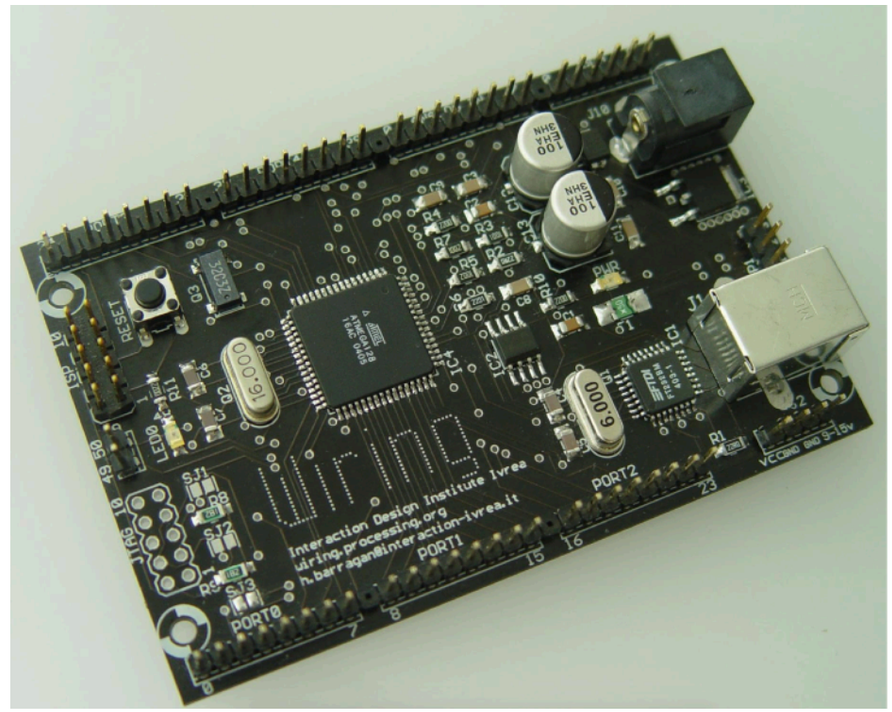
2003

```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

History

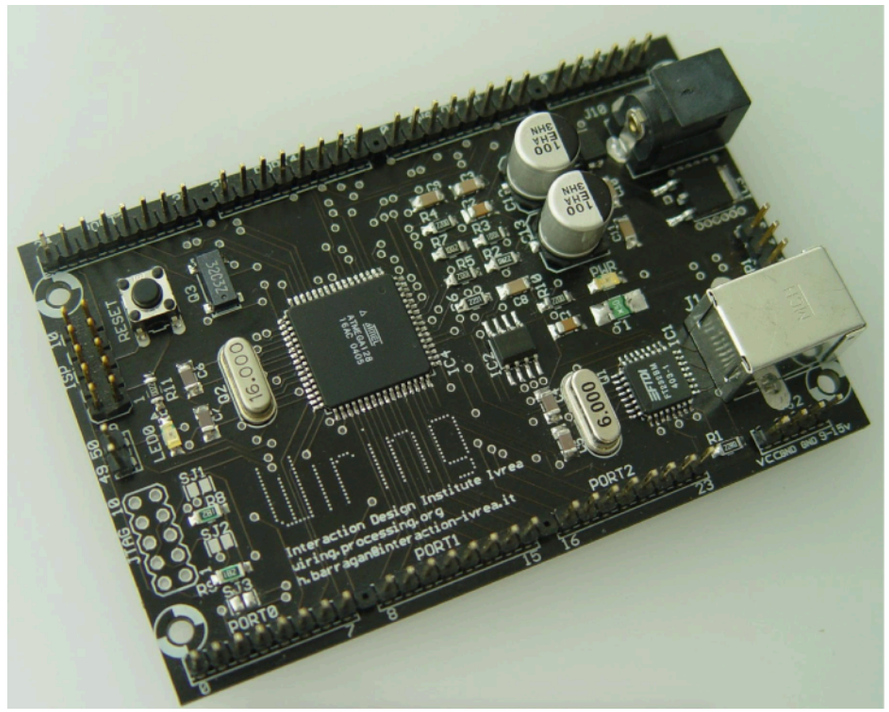
Wiring

2003




```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

History



Wiring

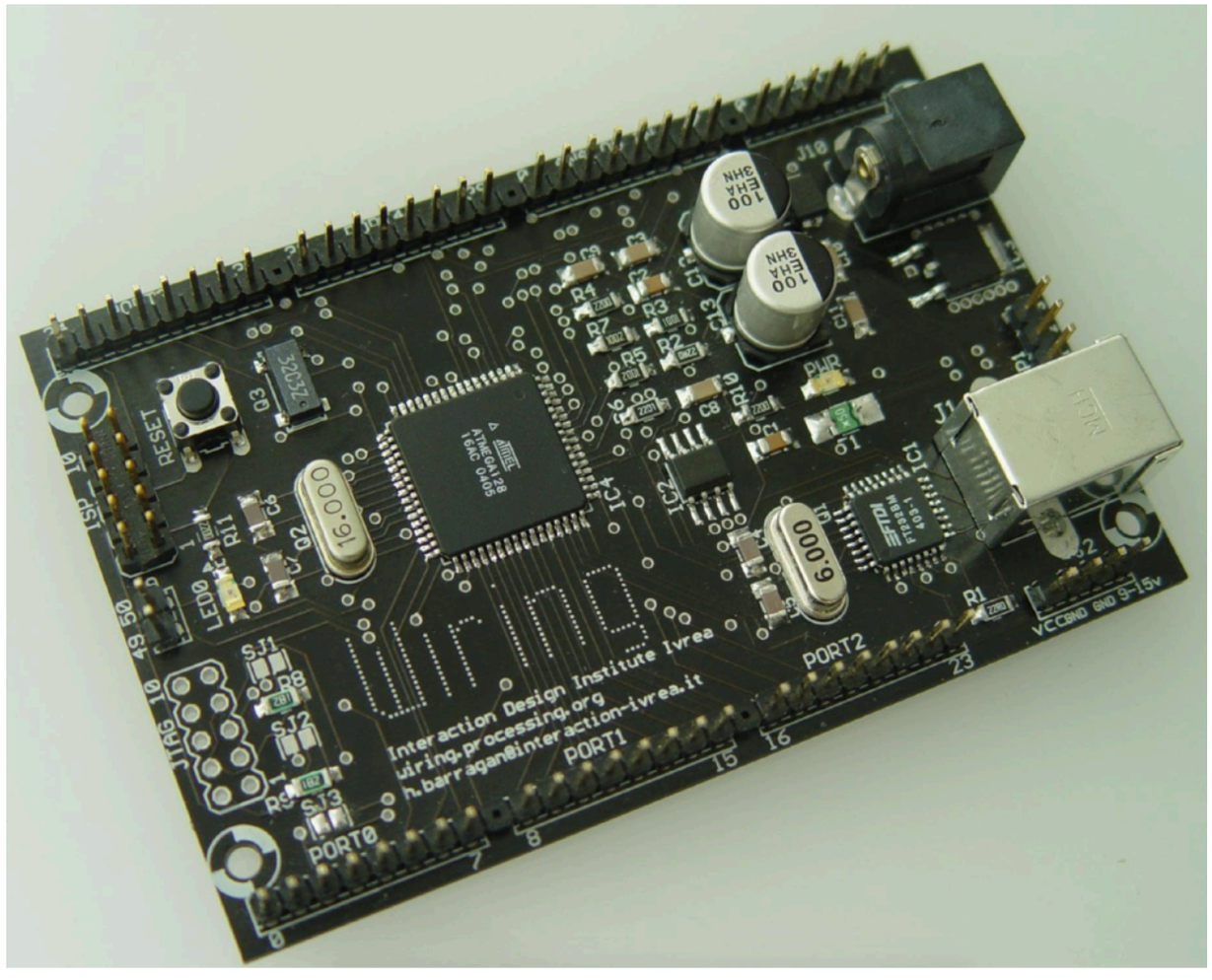
2003


```
int ledPin = WLED;                  // a name for the on-board LED

void setup () {
   pinMode(ledPin, OUTPUT);         // configure the pin for digital output
}

void loop () {
   digitalWrite(ledPin, HIGH);      // turn on the LED
   delay (1000);                    // wait one second (1000 milliseconds)
   digitalWrite(ledPin, LOW);       // turn off the LED
   delay (1000);                    // wait one second
}
```

History

Wiring

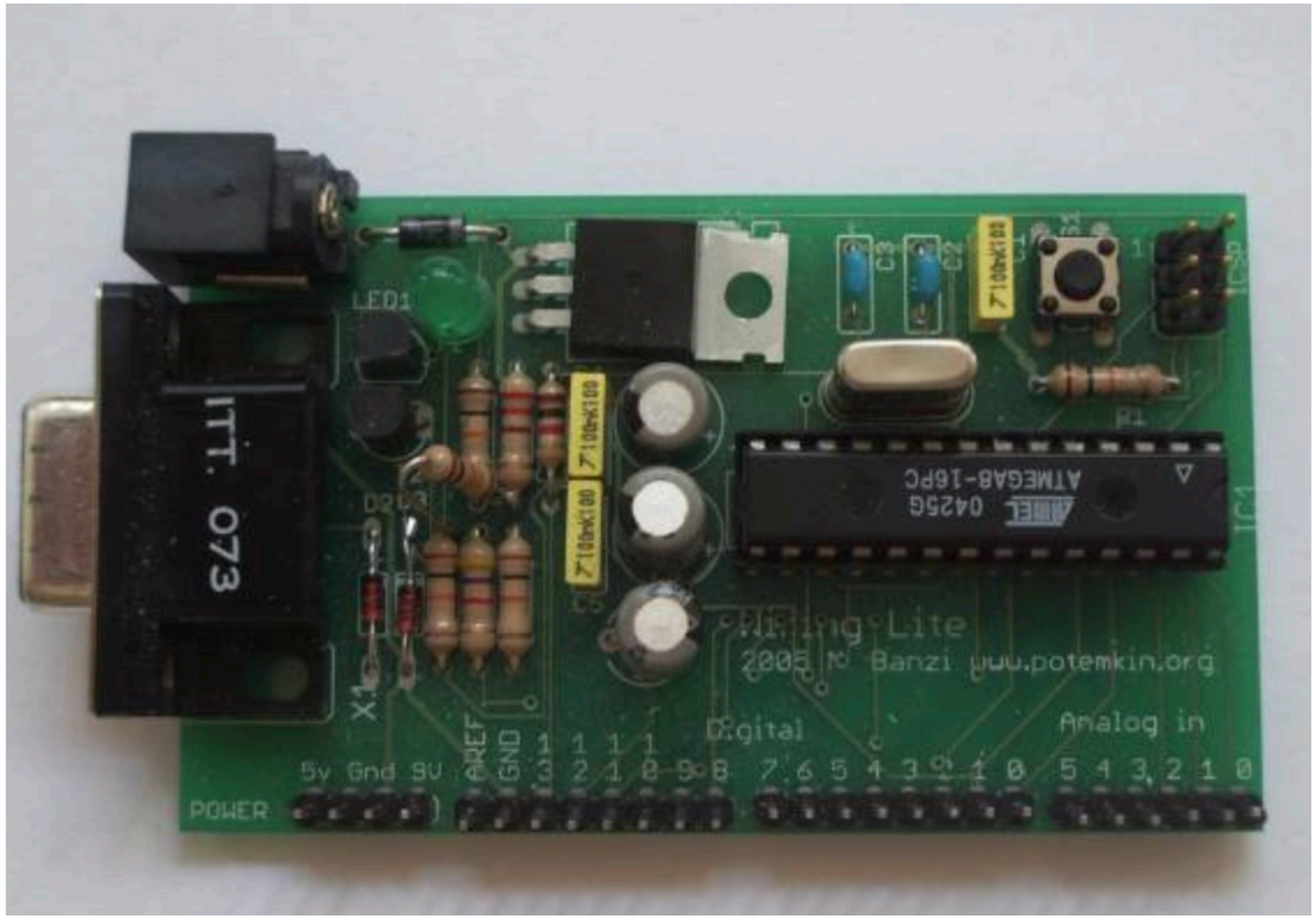
2003



History

Wiring

2003

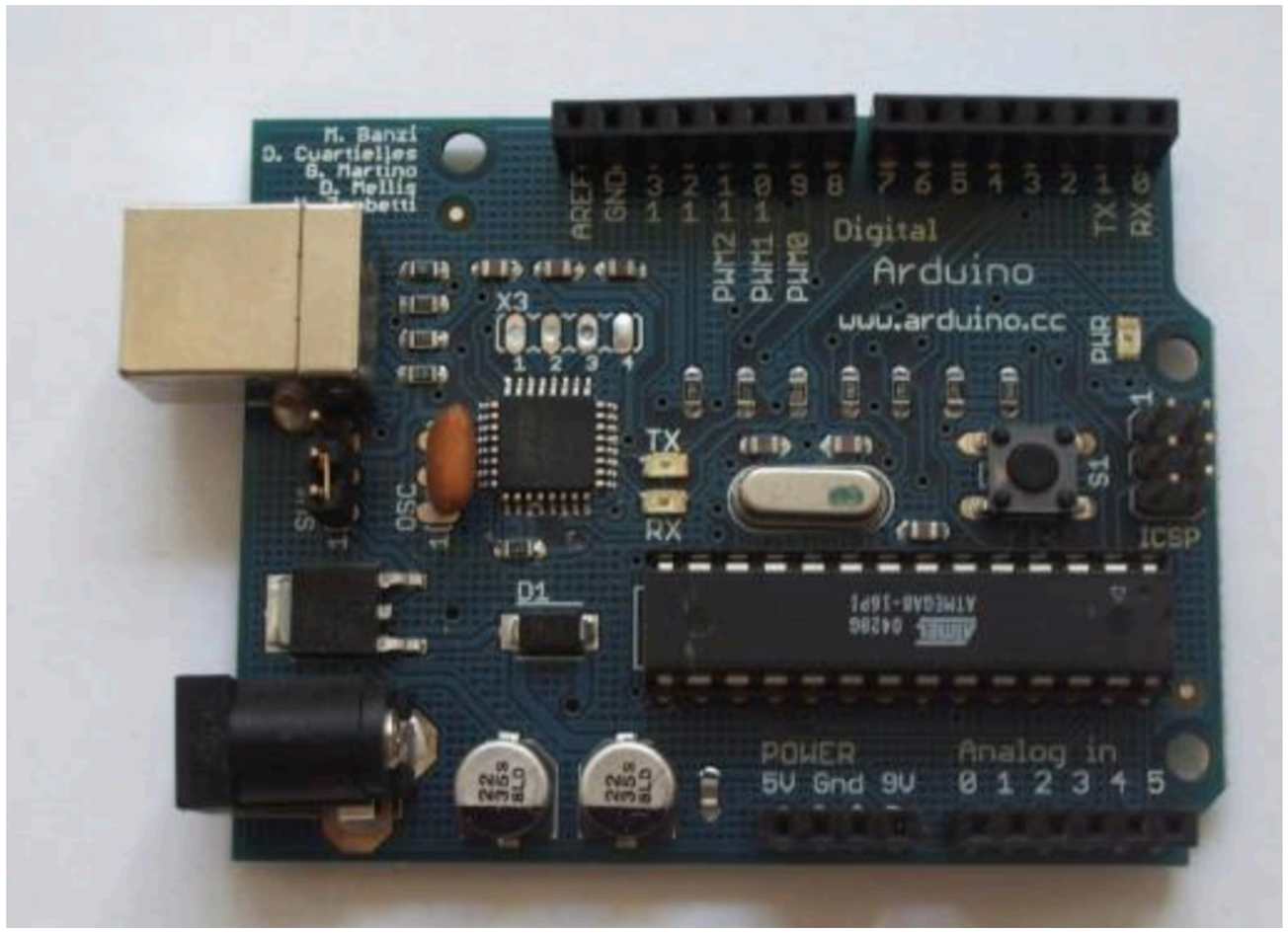


2004

History

Wiring

2003

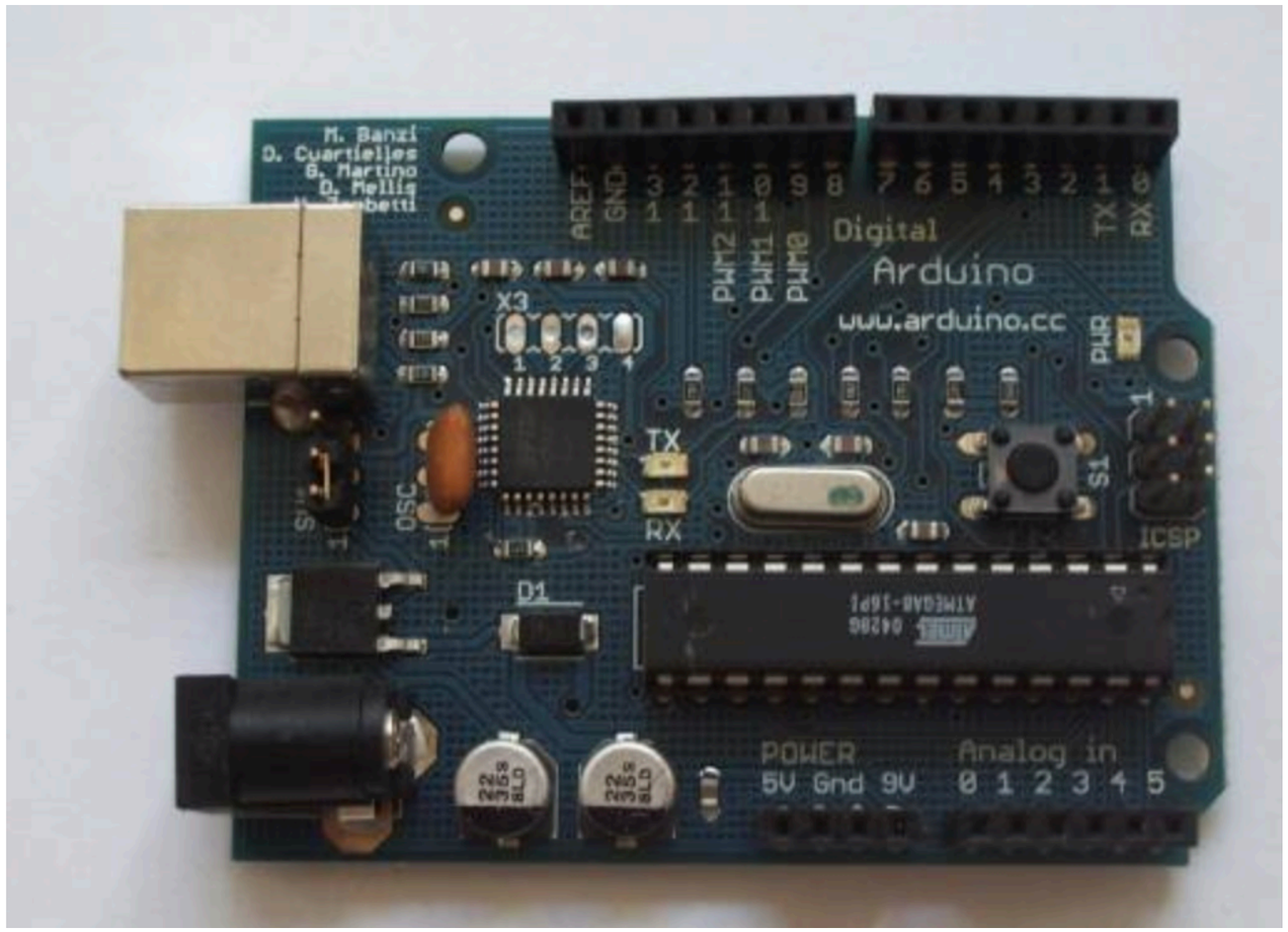


2004

History

Wiring

2003

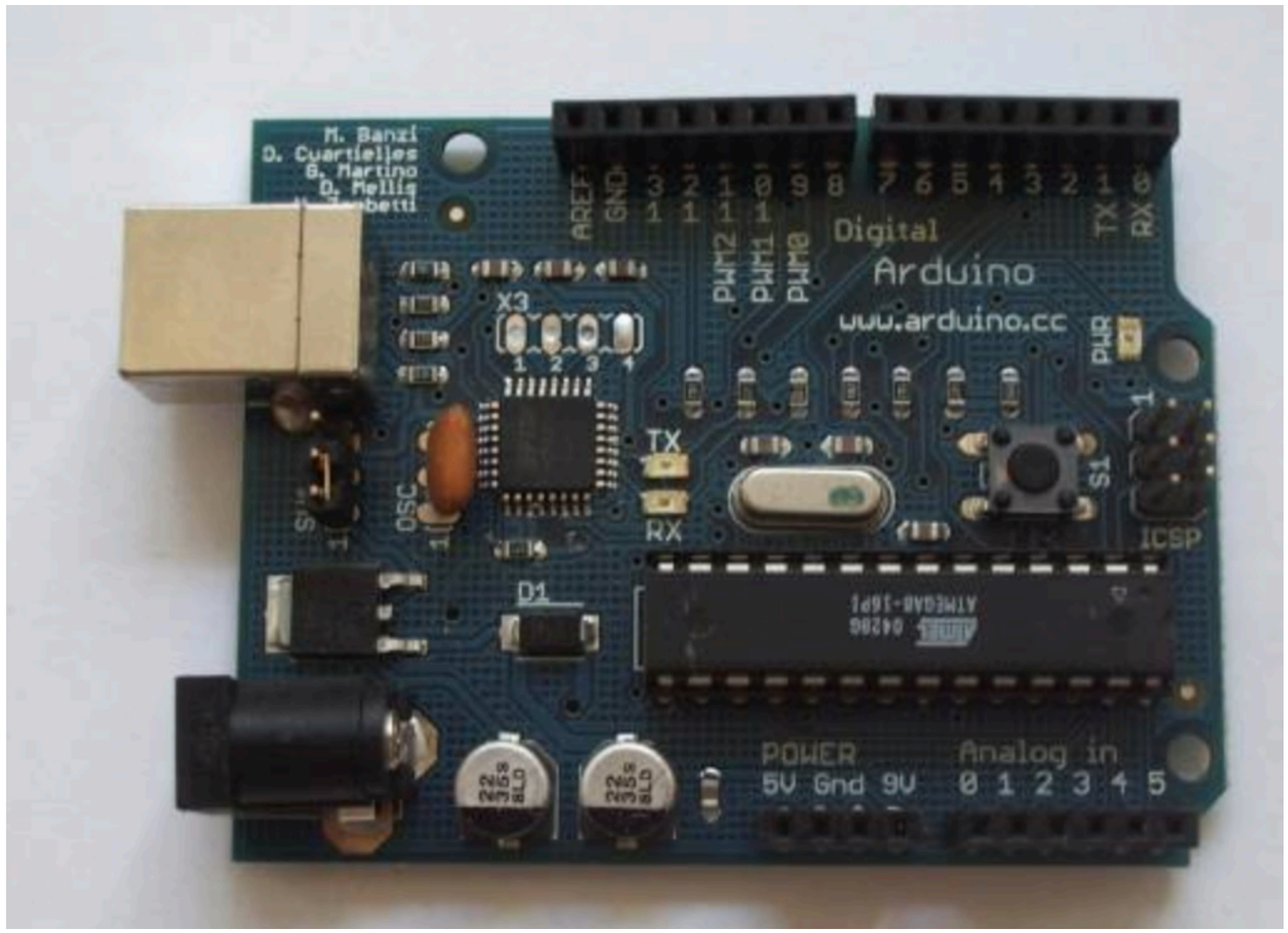


2004

History

Wiring

2003

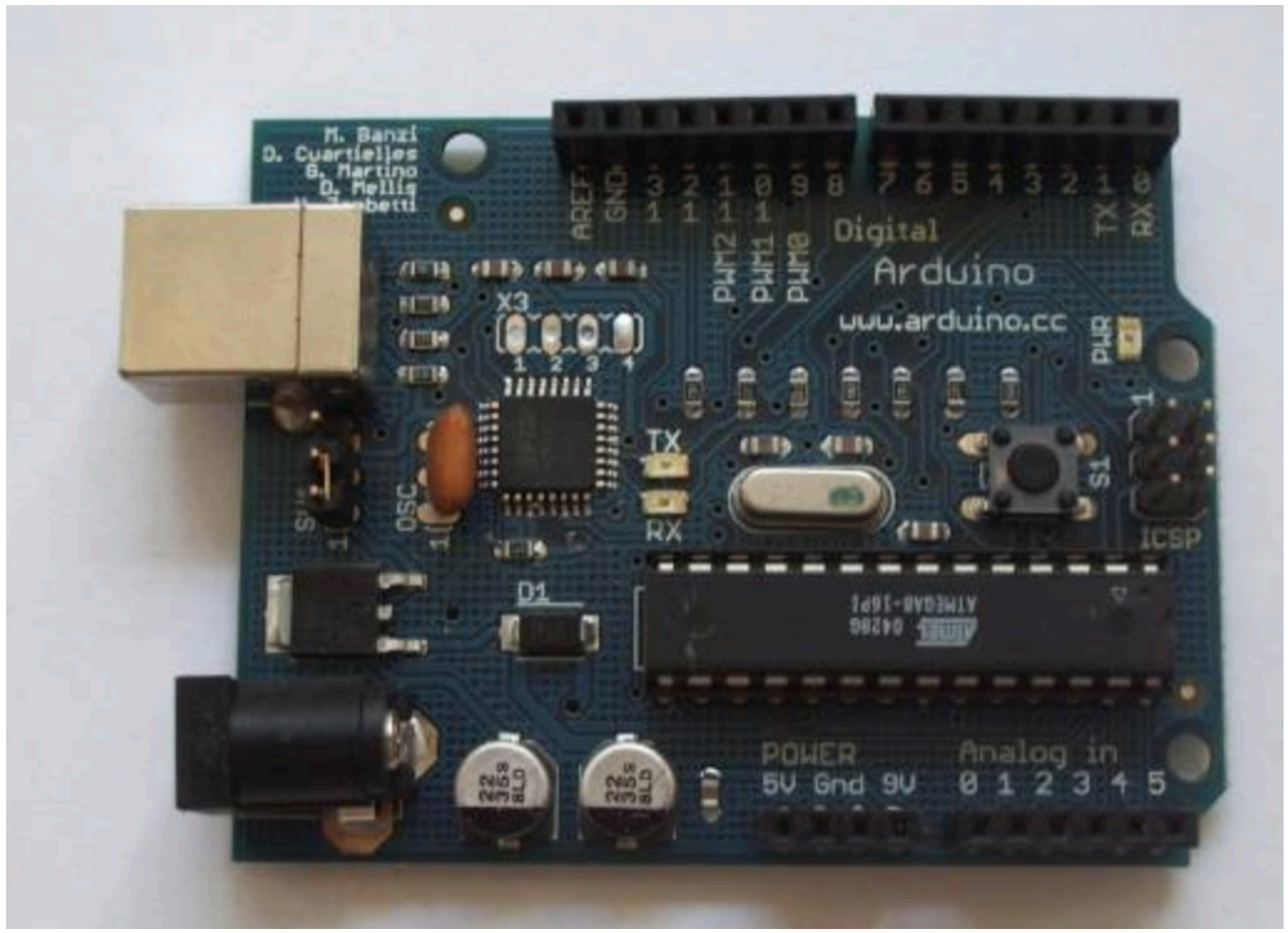


2004 **Arduino**

History

Wiring

2003



2004 **Arduino**

https://arduinohistory.github.io/

Credits

Originally started as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis at the Interaction Design Institute of Ivrea in the early 2000s, it builds upon the Processing project, a language for learning how to code within the context of the visual arts developed by Casey Reas and Ben Fry as well as a thesis project by Hernando Barragan about the Wiring board.

https://www.arduino.cc/en/Main/AboutUs

Credits

Originally started as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis at the Interaction Design Institute of Ivrea in the early 2000s, it builds upon the Processing project, a language for learning how to code within the context of the visual arts developed by Casey Reas and Ben Fry as well as a thesis project by Hernando Barragan about the Wiring board.

https://www.arduino.cc/en/Main/AboutUs

Language

Language

- Functions

Digital I/O

digitalRead()

digitalWrite()

pinMode()

Language

- Functions

Digital I/O

digitalRead()

digitalWrite()

pinMode()

Analog I/O

analogRead()

analogReference()

analogWrite()

Language

- Functions

Digital I/O

digitalRead()

digitalWrite()

pinMode()

Analog I/O

analogRead()

analogReference()

analogWrite()

Time

delay()

delayMicroseconds()

micros()

millis()

Language

- Functions

Digital I/O

digitalRead()

digitalWrite()

pinMode()

Analog I/O

analogRead()

analogReference()

analogWrite()

Zero, Due & MKR Family

analogReadResolution()

analogWriteResolution()

Advanced I/O

noTone()

pulseIn()

pulseInLong()

shiftIn()

shiftOut()

tone()

Time

delay()

delayMicroseconds()

micros()

millis()

Math

abs()

constrain()

map()

max()

min()

pow()

sq()

sqrt()

Trigonometry

cos()

sin()

tan()

Characters

isAlpha()

isAlphaNumeric()

isAscii()

isControl()

isDigit()

isGraph()

isHexadecimalDigit()

isLowerCase()

isPrintable()

isPunct()

isSpace()

isUpperCase()

isWhitespace()

Random Numbers

random()

randomSeed()

Bits and Bytes

bit()

bitClear()

bitRead()

bitSet()

bitWrite()

highByte()

lowByte()

External Interrupts

attachInterrupt()

detachInterrupt()

Interrupts

interrupts()

noInterrupts()

Communication

Serial

Stream

USB

Keyboard

Mouse

Language

- Functions
- Variables

Constants

Floating Point Constants

Integer Constants

HIGH | LOW

INPUT | OUTPUT | INPUT_PULLUP

LED_BUILTIN

true | false

Conversion

byte()

char()

float()

int()

long()

word()

Data Types

String()

array

bool

boolean

byte

char

double

float

int

long

short

size_t

string

unsigned char

unsigned int

unsigned long

void

word

Variable Scope & Qualifiers

const

scope

static

volatile

Utilities

PROGMEM

sizeof()

Language

- Functions
- Variables
- Structure

Sketch

loop()

setup()

Language

- Functions
- Variables
- Structure

Sketch

loop()

setup()

Control Structure

break

continue

do...while

else

for

goto

if

return

switch...case

while

Language

- Functions
- Variables
- Structure

Sketch

loop()

setup()

Control Structure

break

continue

do...while

else

for

goto

if

return

switch...case

while

Further Syntax

#define (define)

#include (include)

/* */ (block comment)

// (single line comment)

; (semicolon)

{} (curly braces)

Arithmetic Operators

% (remainder)

* (multiplication)

+ (addition)

- (subtraction)

/ (division)

= (assignment operator)

Comparison Operators

!= (not equal to)

< (less than)

<= (less than or equal to)

== (equal to)

> (greater than)

>= (greater than or equal t

Boolean Operators

! (logical not)

&& (logical and)

|| (logical or)

Pointer Access Operators

& (reference operator)

* (dereference operator)

Bitwise Operators

& (bitwise and)

<< (bitshift left)

>> (bitshift right)

^ (bitwise xor)

| (bitwise or)

~ (bitwise not)

Compound Operators

%= (compound remainder)

&= (compound bitwise and)

*= (compound multiplication)

++ (increment)

+= (compound addition)

-- (decrement)

-= (compound subtraction)

/= (compound division)

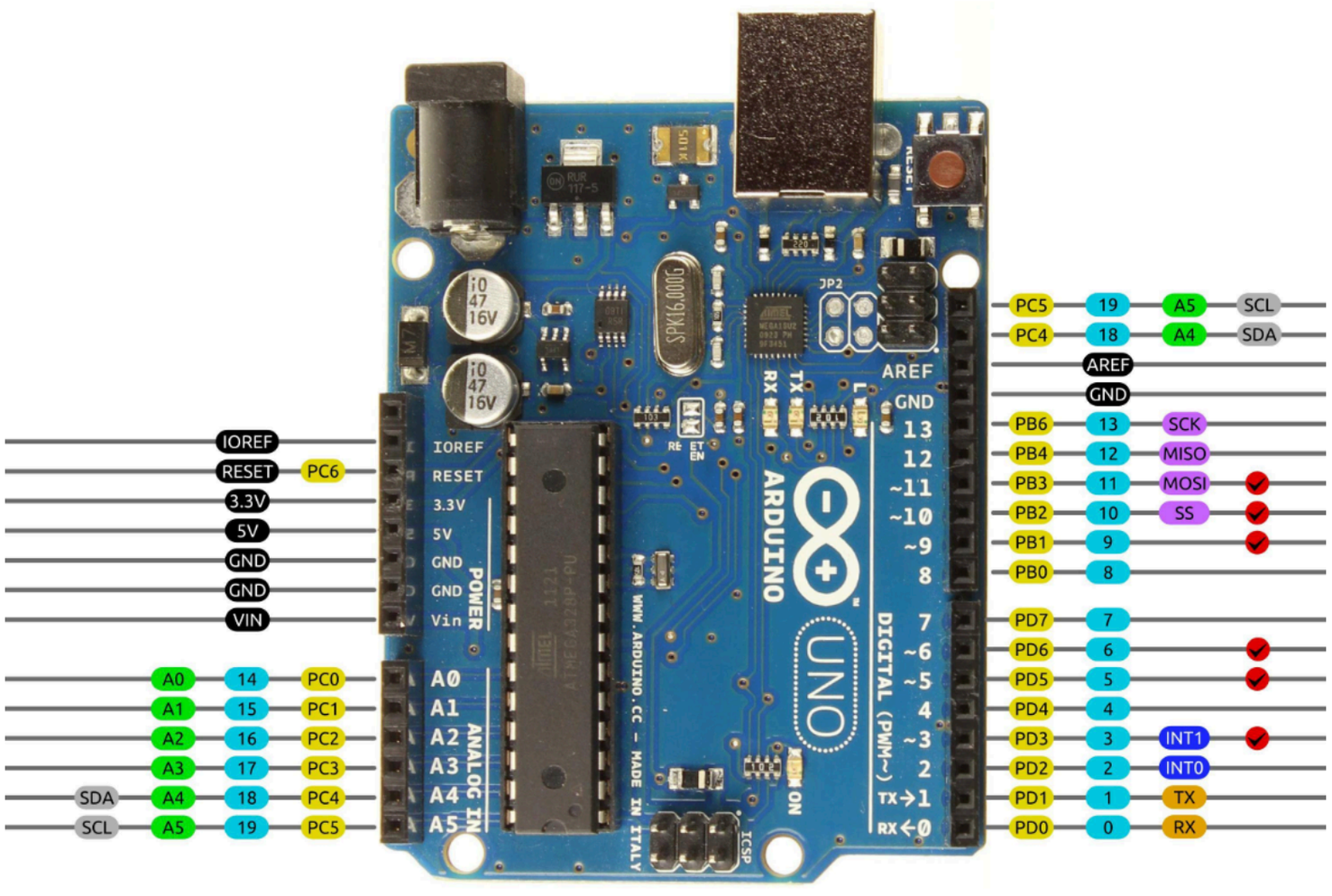
^= (compound bitwise xor)

|= (compound bitwise or)

https://www.arduino.cc/reference/en/

Digital Pins

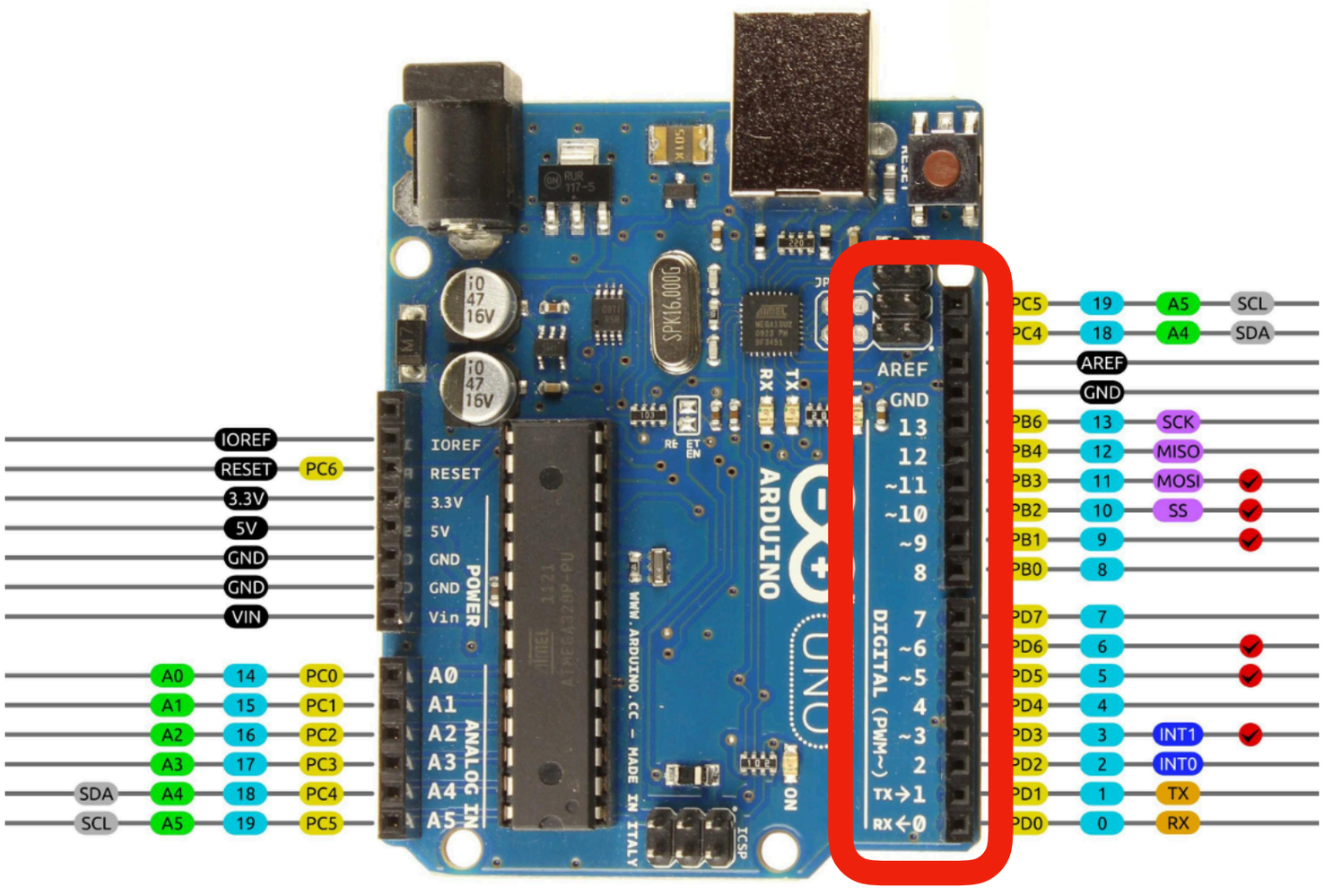
Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Digital Pins

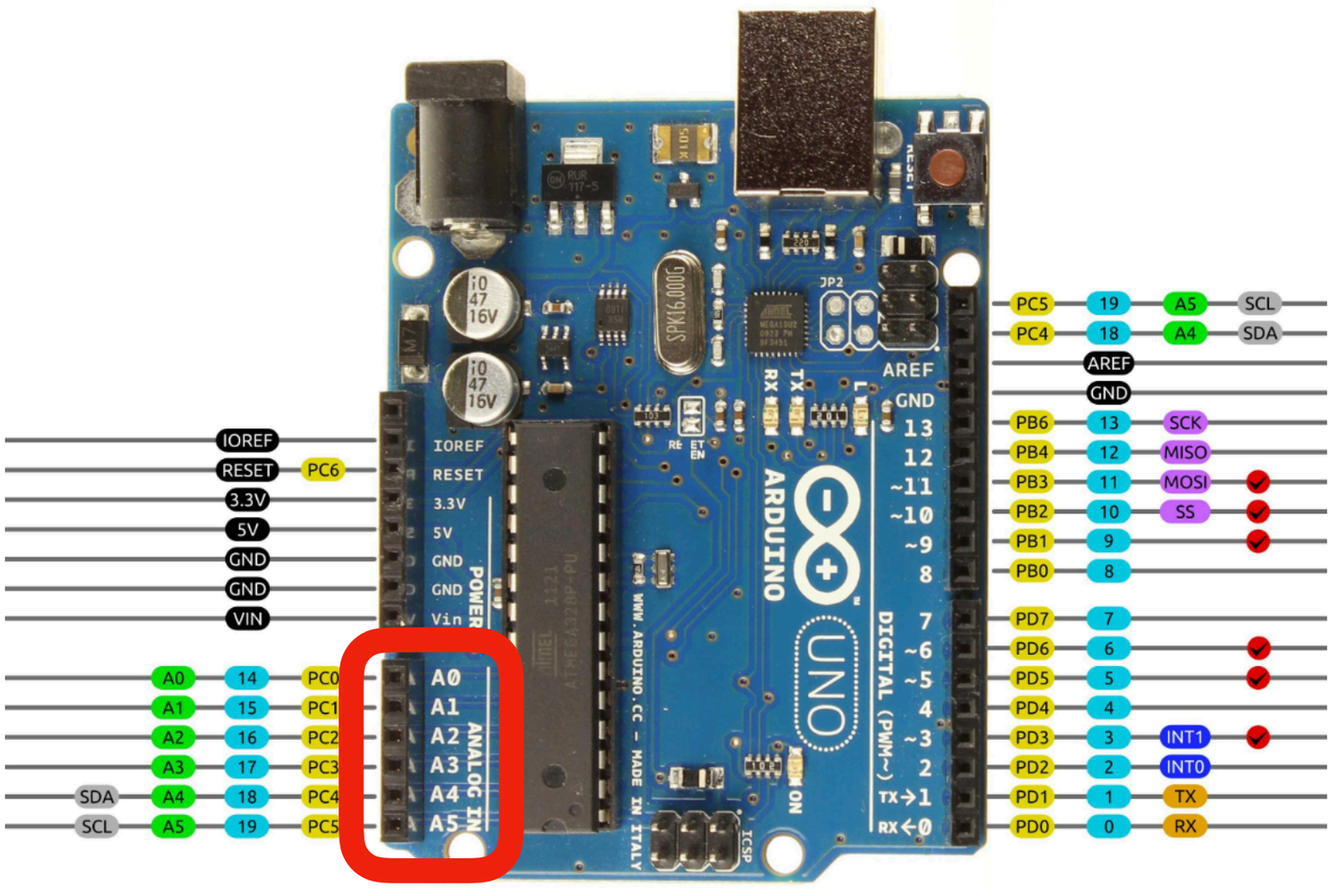
Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Analog Pins

Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Accessing the pins

```
void setup() {
  pinMode(13, OUTPUT);    // sets the digital pin 13 as output
}

void loop() {
  digitalWrite(13, HIGH); // sets the digital pin 13 on
  delay(1000);            // waits for a second
  digitalWrite(13, LOW);  // sets the digital pin 13 off
  delay(1000);            // waits for a second
}
```

Accessing the pins

```
int ledPin = 13;  // LED connected to digital pin 13
int inPin = 7;    // pushbutton connected to digital pin 7
int val = 0;      // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT);  // sets the digital pin 13 as output
  pinMode(inPin, INPUT);    // sets the digital pin 7 as input
}

void loop() {
  val = digitalRead(inPin);   // read the input pin
  digitalWrite(ledPin, val);  // sets the LED to the button's value
}
```

Accessing the pins

```
int analogPin = A3; // potentiometer wiper (middle terminal) connected to analog pin 3
                    // outside leads to ground and +5V
int val = 0;  // variable to store the value read

void setup() {
  Serial.begin(9600);           //  setup serial
}

void loop() {
  val = analogRead(analogPin);  // read the input pin
  Serial.println(val);          // debug value
}
```

Accessing the pins

```
int ledPin = 9;      // LED connected to digital pin 9
int analogPin = 3;   // potentiometer connected to analog pin 3
int val = 0;         // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT);  // sets the pin as output
}

void loop() {
  val = analogRead(analogPin);  // read the input pin
  analogWrite(ledPin, val / 4);
  // analogRead values go from 0 to 1023, analogWrite values from 0 to 255
}
```

Connect to Android Things

```
private val uart: UartDevice by lazy {
    PeripheralManager.getInstance().openUartDevice(uartDevice).apply {
        setBaudrate(115200)
        setDataSize(8)
        setParity(UartDevice.PARITY_NONE)
        setStopBits(1)
    }
}
```

Connect to Android Things

```
fun read(): String {
    val maxCount = 8
    val buffer = ByteArray(maxCount)
    var output = ""
    do {
        val count = uart.read(buffer, buffer.size)
        output += buffer.toReadableString()
        if (count == 0) break
        logd("Read ${buffer.toReadableString()} $count bytes from
peripheral")
    } while (true)
    return output
}


private fun ByteArray.toReadableString() = filter { it > 0.toByte() }
    .joinToString(separator = "") { it.toChar().toString() }
```

```
fun read(): String {
    val maxCount = 8
    val buffer = ByteArray(maxCount)
    var output = ""
    do {
        val count = uart.read(buffer, buffer.size)
        output += buffer.toReadableString()
        if (count == 0) break
        logd("Read ${buffer.toReadableString()} $count bytes from
peripheral")
    } while (true)
    return output
}


private fun ByteArray.toReadableString() = filter { it > 0.toByte() }
    .joinToString(separator = "") { it.toChar().toString() }

fun write(value: String) {
    val count = uart.write(value.toByteArray(), value.length)
    logd("Wrote $value $count bytes to peripheral")
}
```

Connect to Android Things

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  processSerialCommand();
}
```

Connect to Android Things

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  processSerialCommand();
}
```

```
void processSerialCommand() {
  if (Serial.available() > 0) {
    char command = (char) Serial.read();
    switch (command) {
      case 'X':
        doStuff();
        break;
    }
    Serial.flush();
  }
}
```

Connect to Android Things

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  processSerialCommand();
}
```

```
void processSerialCommand() {
  if (Serial.available() > 0) {
    char command = (char) Serial.read();
    switch (command) {
      case 'X':
        doStuff();
        break;
    }
    Serial.flush();
  }
}
```

```
void doStuff() {
  int sensorValue = analogRead(A0);
  sprintf(buff, "%d", sensorValue);
  Serial.write(buff);
}
```

Lecture outcomes

- Using Arduino platform.
- Link Android Things boards with Arduino.
- Capture sensors data.