https://www.halvorsen.blog

Arduino UNO R4

Arduino UNO R4 Minima and Arduino UNO R4 WiFi

Hans-Petter Halvorsen

Contents

https://www.halvorsen.blog

Introduction to Arduino UNO R4

Hans-Petter Halvorsen

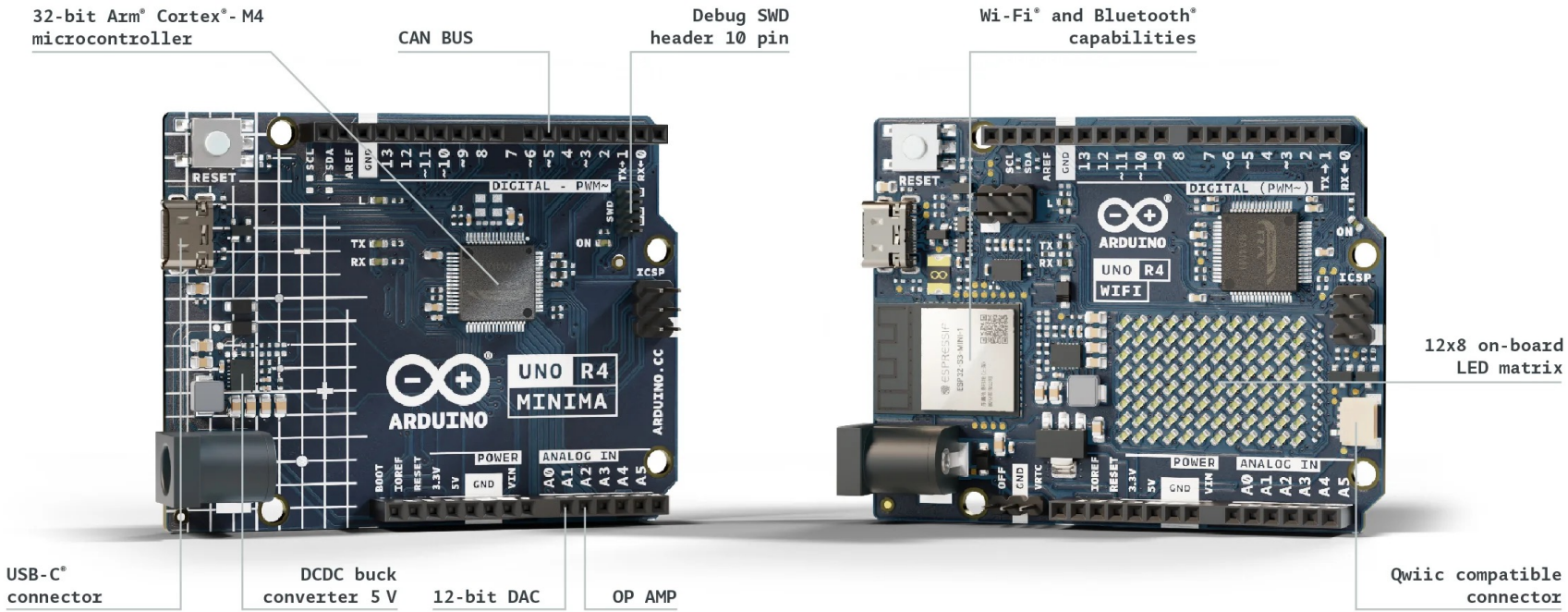
Table of Contents

Arduino UNO R4

- In mid 2023 a new version of the popular Arduino UNO R3 was released
- Arduino UNO R4 comes in 2 different versions:
  - Arduino UNO R4 Minima (about €18)
  - Arduino UNO R4 WiFi (about €25)

Arduino UNO R4 Minima vs WiFi

https://store.arduino.cc/pages/uno-r4



If you compare the feature list and the price difference between Minima and the WiFi edition, I see no reasons why you should not buy the WiFi edition

Arduino UNO R3 vs R4

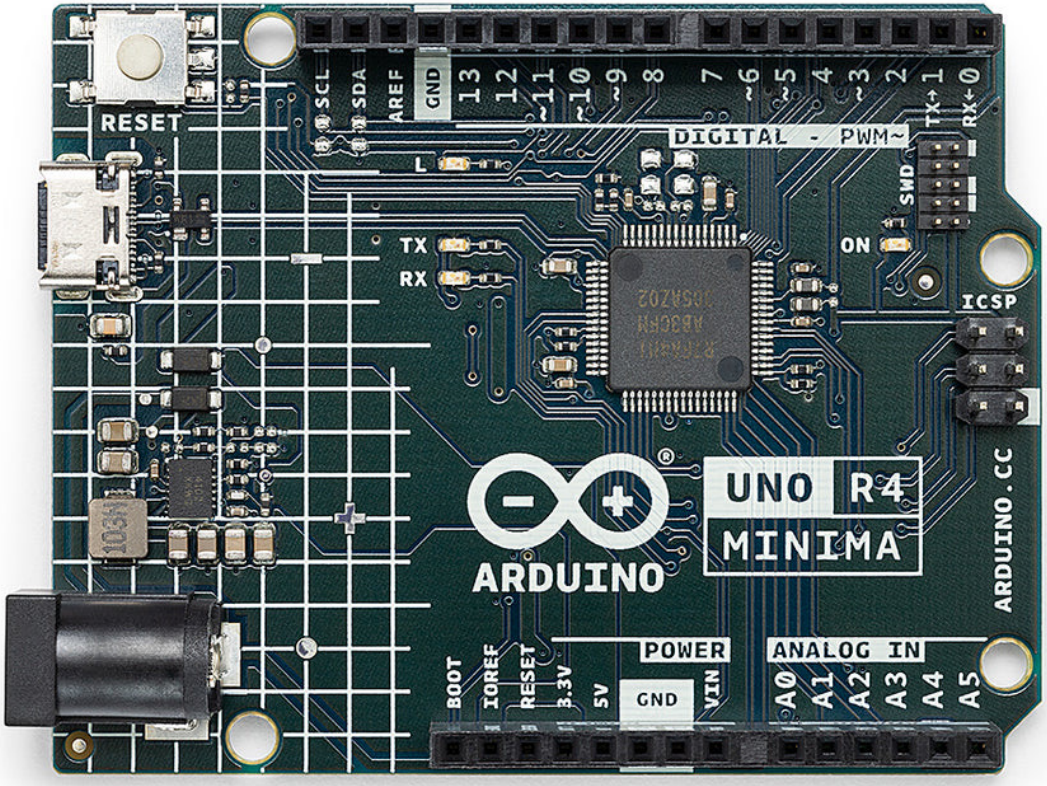
Parameter	Arduino UNO R3	Arduino UNO R4
Microcontroller (CPU)	8-bit	32-bit
WiFi	No	Arduino UNO R4 WiFi
Bluetooth	NO	Arduino UNO R4 WiFi
Memory	2KB SRAM, 32KB FLASH	256 kB Flash, 32 kB RAM
Input voltage (VIN)	6-20 V	6-24 V
Digital I/O pins	14	14
Analog input pins	6(10-bit)	6(14-bit)
PWM pins	6(12-bit)	6(12-bit)
DAC pin	0	1
USN port	USB-B	USB-C
CAN bus	0	1
Qwiic connector	No	Arduino UNO R4 WiFi
LED matrix	No	12x8 LED matrix on Arduino UNO R4 WiFi
Maximum Pin Current	20mA	8mA

Improvements Arduino UNO R3 vs R4

Main differences:

- 32-bit CPU instead of 8-bit CPU
- 256kB flash memory instead 32kB
- 32 kB RAM instead of 2kB SRAM
- USB-C instead of USB-B port
- 12-bit DAC, while R3 has only PWM
- …

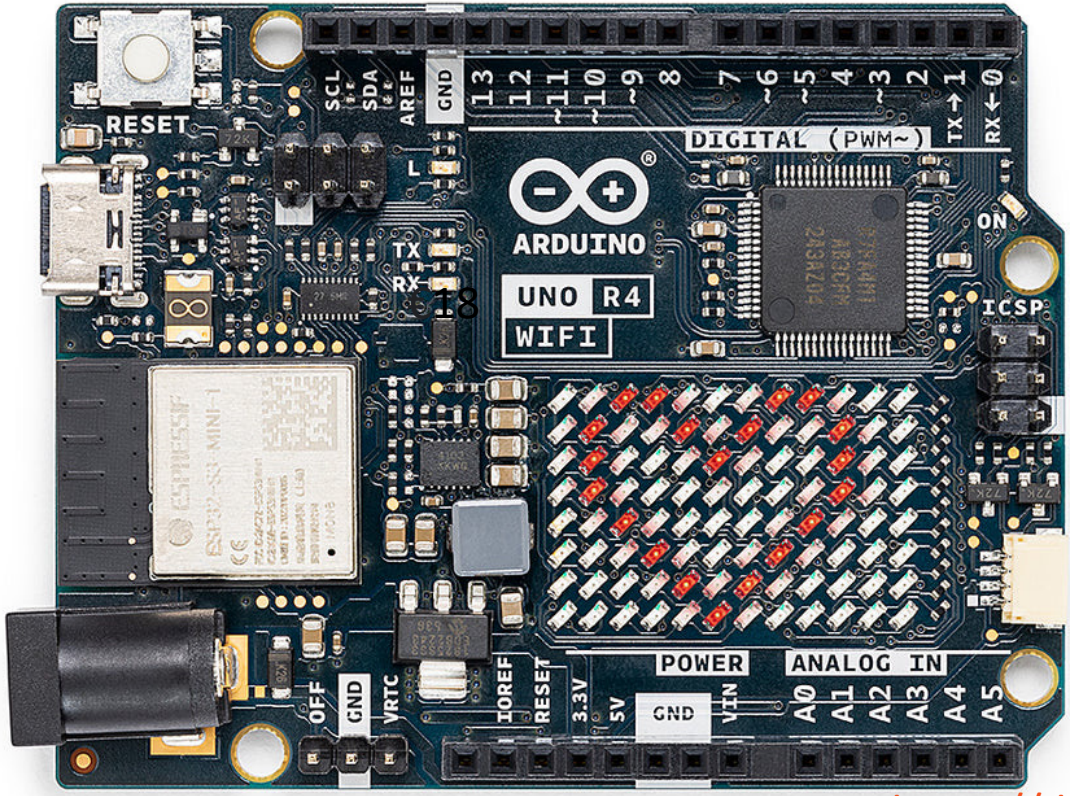
Arduino UNO R4 Minima



Board	**Name**	Arduino® UNO R4 Minima
	SKU	ABX00080
Microcontroller	Renesas RA4M1 (Arm® Cortex®-M4)	
USB	**USB-C®**	Programming Port
Pins	**Digital I/O Pins**	14
	Analog input pins	6
	DAC	1
	PWM pins	6
Communication	**UART**	Yes, 1x
	I2C	Yes, 1x
	SPI	Yes, 1x
	CAN	Yes 1 CAN Bus
Power	**Circuit operating voltage**	5 V
	Input voltage (VIN)	6-24 V
	DC Current per I/O Pin	8 mA
Clock speed	**Main core**	48 MHz
Memory	**RA4M1**	256 kB Flash, 32 kB RAM
Dimensions	**Width**	68.85 mm
	Length	53.34 mm

https://docs.arduino.cc/hardware/uno-r4-minima

Arduino UNO R4 WiFi



- WiFi
- Bluetooth
- LED Matrix

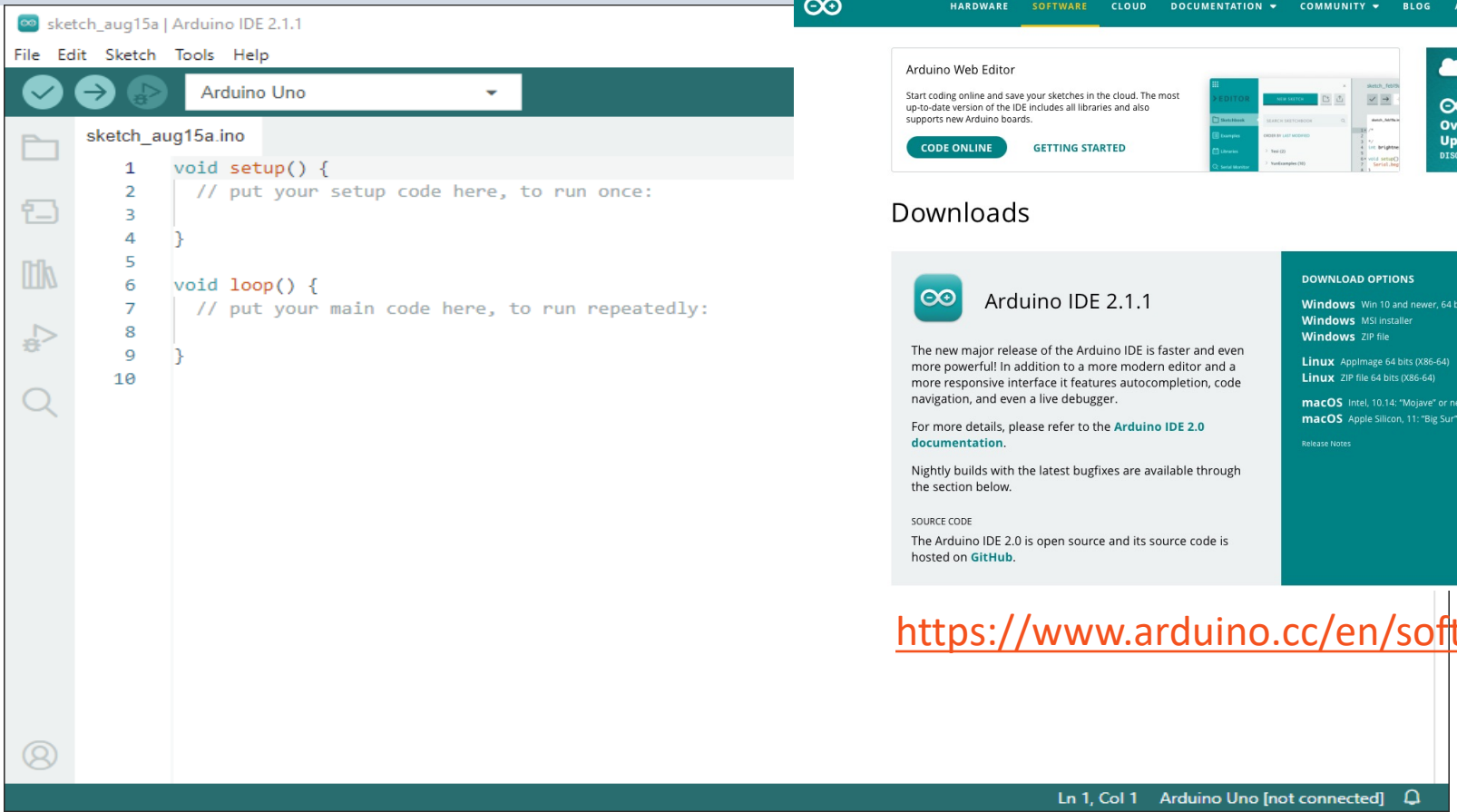
https://docs.arduino.cc/hardware/uno-r4-wifi

Arduino IDE

Hans-Petter Halvorsen

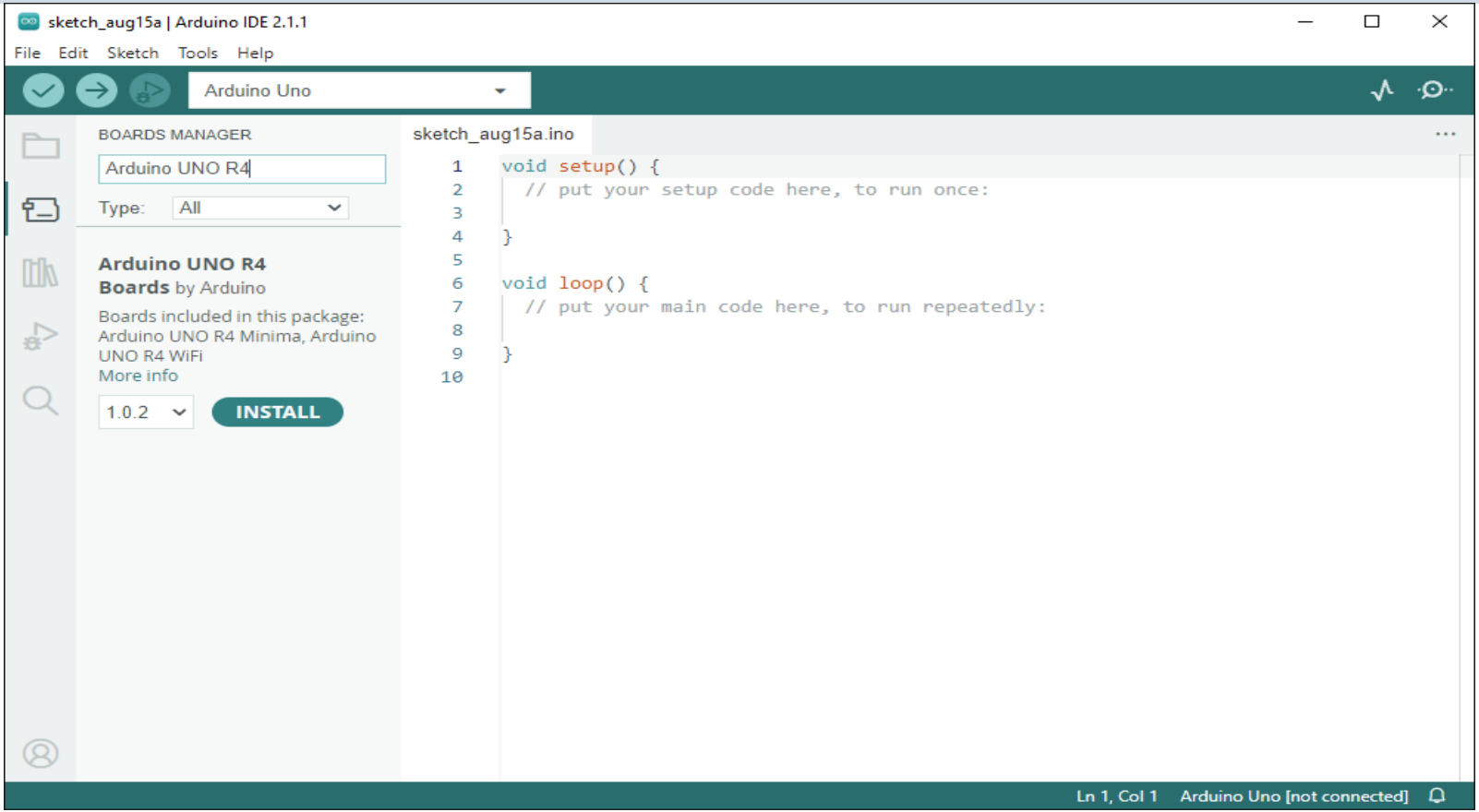
Table of Contents

Arduino IDE

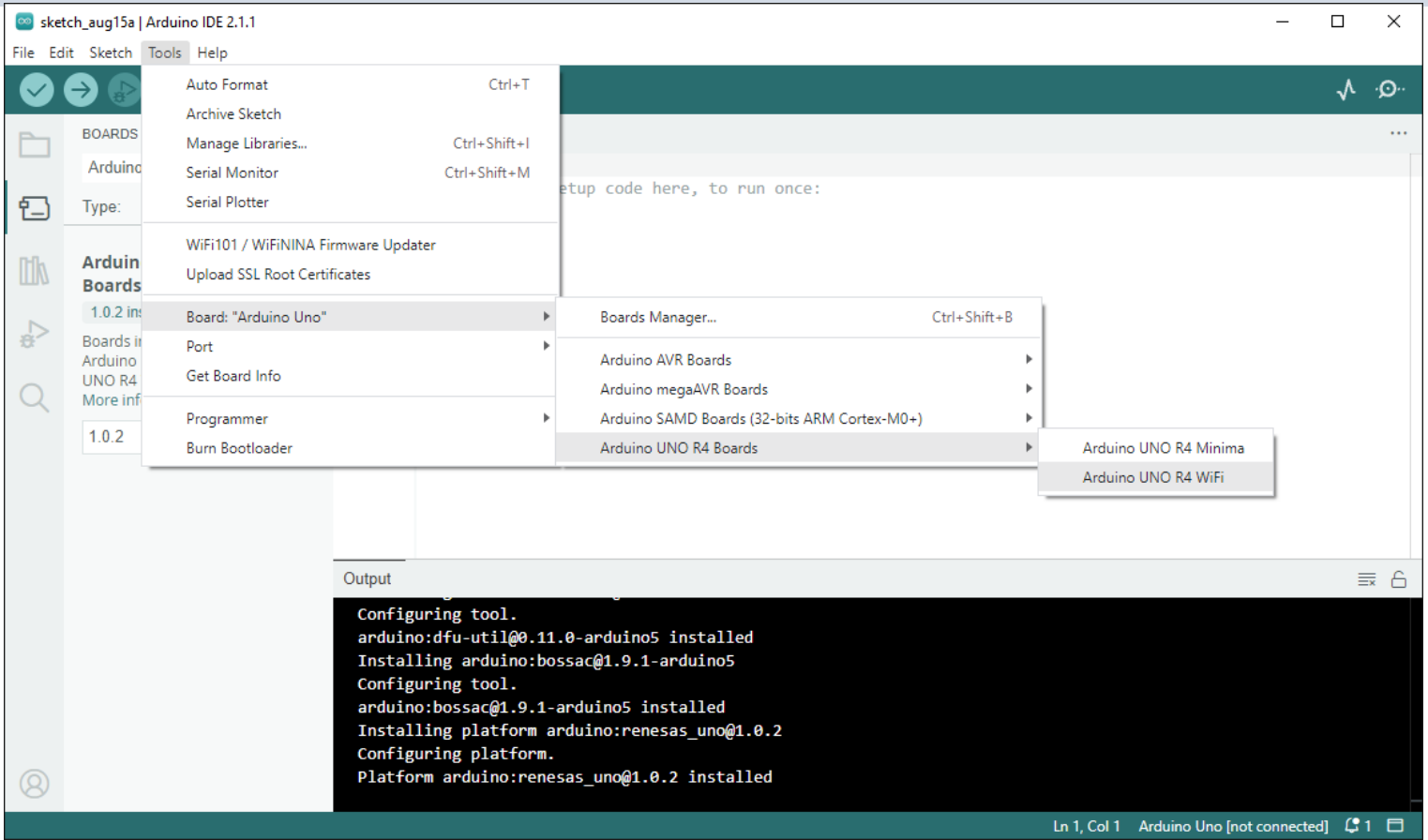


https://www.arduino.cc/en/software

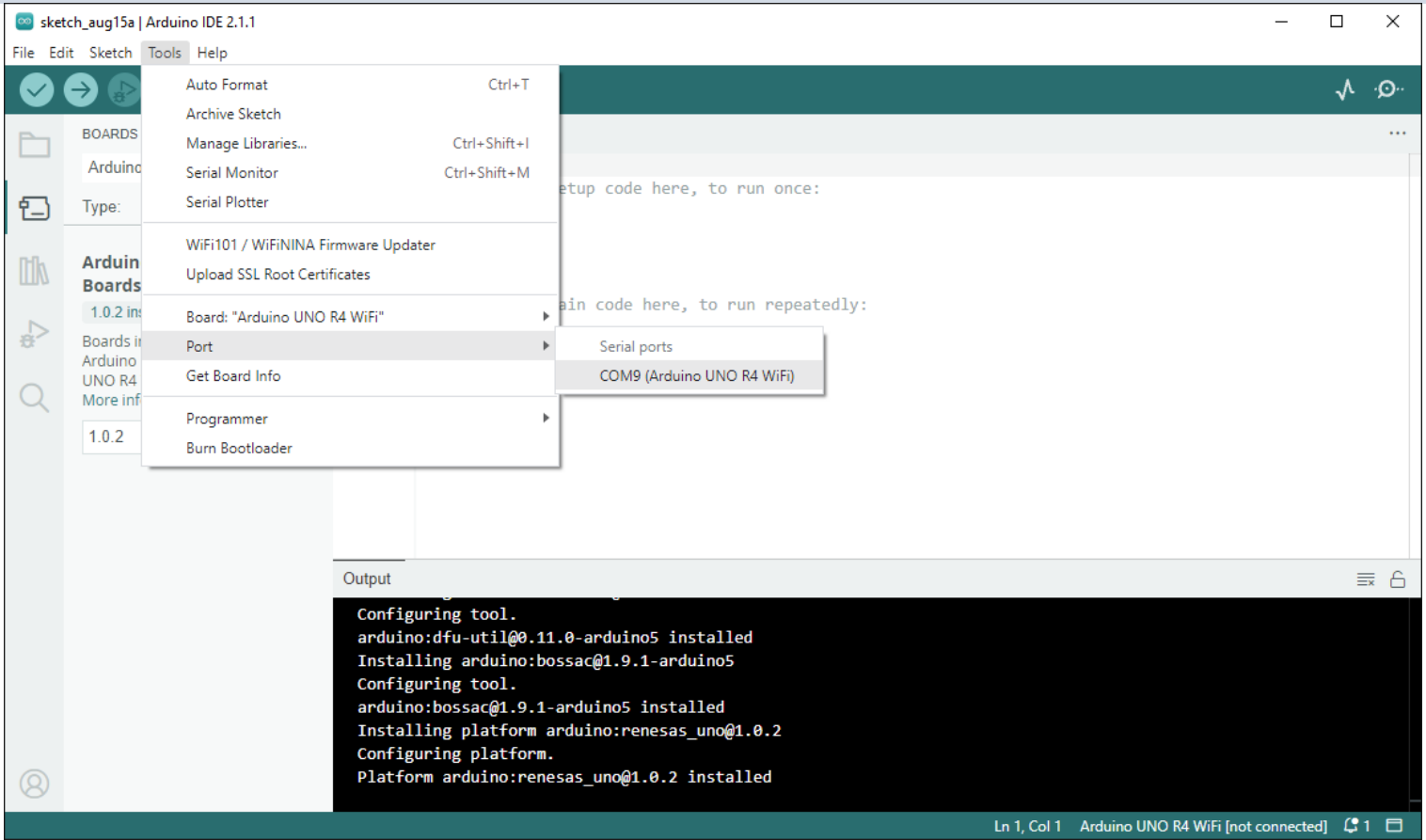
Install Arduino UNO R4 Board



Select Arduino UNO R4 Board



Select Port


sketch_aug15a | Arduino IDE 2.1.1
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "Arduino UNO R4 WiFi"
Port
Get Board Info
Programmer
Burn Bootloader
Serial ports
COM9 (Arduino UNO R4 WiFi)
BOARDS
Type:
Output
Configuring tool.
arduino:dfu-util@0.11.0-arduino5 installed
Installing arduino:bossac@1.9.1-arduino5
Configuring tool.
arduino:bossac@1.9.1-arduino5 installed
Installing platform arduino:renesas_uno@1.0.2
Configuring platform.
Platform arduino:renesas_uno@1.0.2 installed
Ln 1, Col 1
Arduino UNO R4 WiFi [not connected]

Start creating Arduino Programs

All Arduino programs must follow the following main structure:

```
// Initialization, define variables, etc.

void setup()
{
      // Initialization
      ...
}

void loop()
{
      //Main Program
      ...
}
```

Built-in LED Example

blinking_led_builtin | Arduino IDE 2.1.1

File Edit Sketch Tools Help

Arduino UNO R4 WiFi

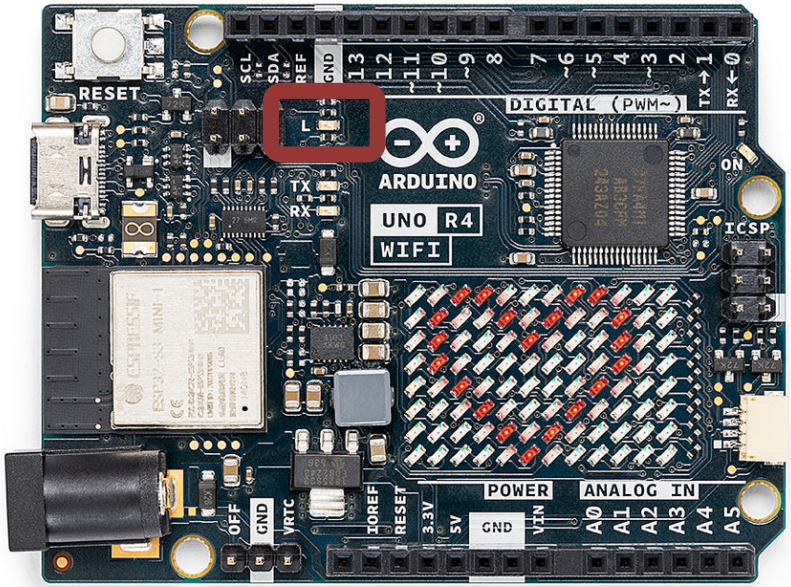
blinking_led_builtin.ino

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

Output

```
[==========                    ] 33% (3/9 pages)
[=============                 ] 44% (4/9 pages)
[================              ] 55% (5/9 pages)
[====================          ] 66% (6/9 pages)
[========================      ] 77% (7/9 pages)
[============================  ] 88% (8/9 pages)
[==============================] 100% (9/9 pages)
Done in 2.300 seconds
```

Building sketch Ln 1, Col 14 Arduino UNO R4 WiFi on COM9



Built-in LED Example

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

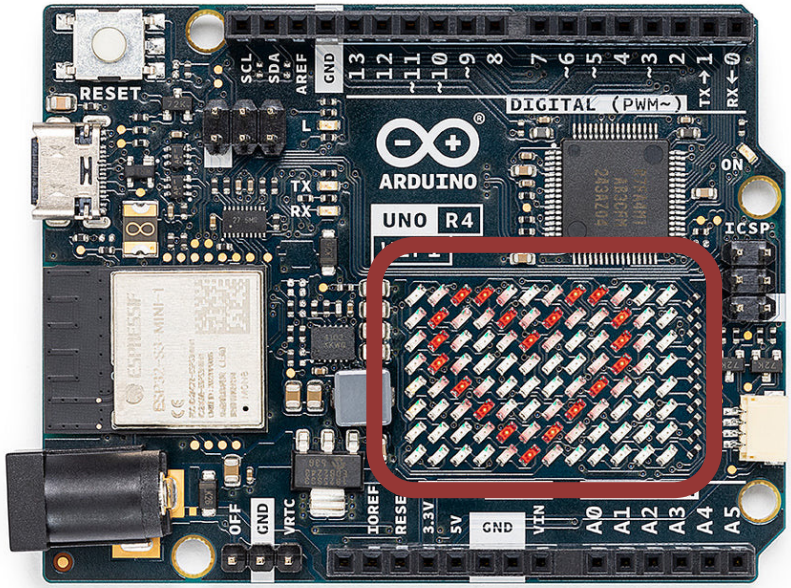
LED Matrix

Hans-Petter Halvorsen

Table of Contents

LED Matrix (Arduino UNO R4 WiFi)

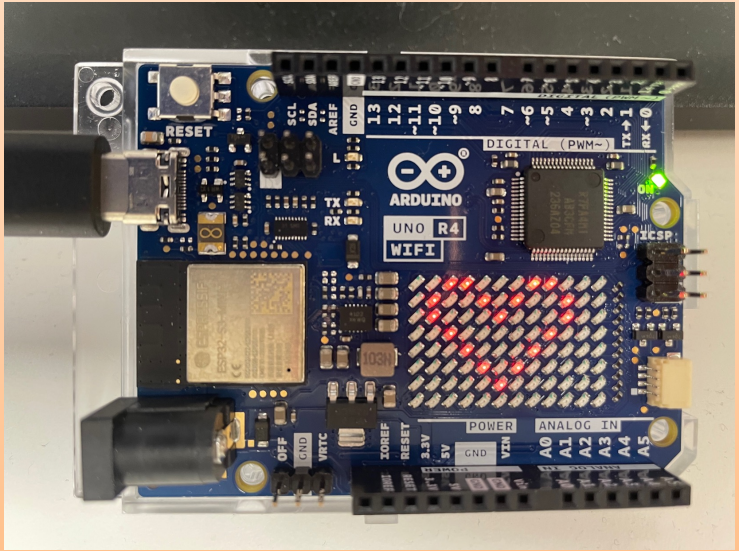
The Arduino UNO R4 WiFi comes with a built in **12x8** LED Matrix, that is available to be programmed to display graphics, animations, act as an interface, or even play games on (e.g., Tetris or Snake?).



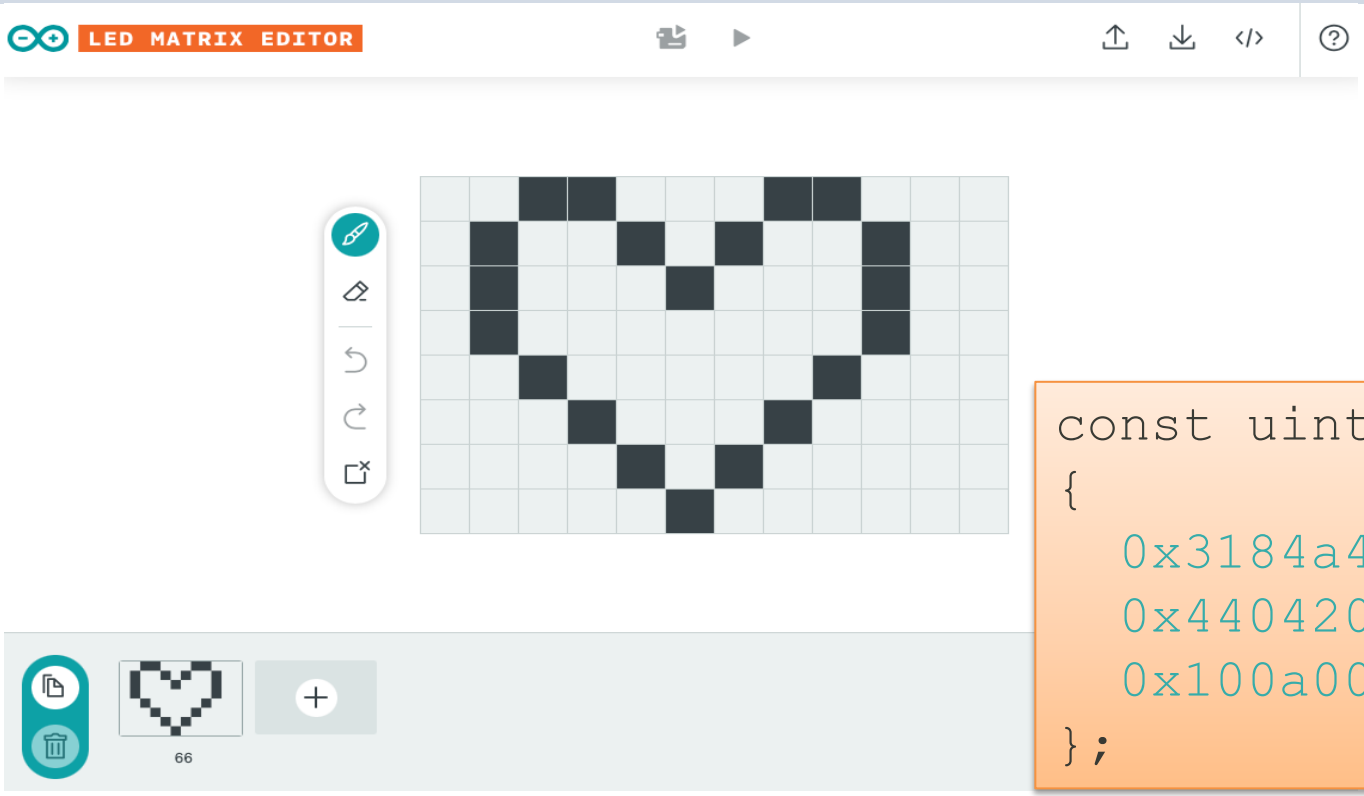
https://docs.arduino.cc/tutorials/uno-r4-wifi/led-matrix

```
#include "Arduino_LED_Matrix.h"

uint8_t heart[8][12] = {
  { 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0 },
  { 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0 },
  { 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0 },
  { 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0 },
  { 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
};

ArduinoLEDMatrix matrix;

void setup()
{
  Serial.begin(115200);
  matrix.begin();
  matrix.renderBitmap(heart, 8, 12);
}

void loop()
{

}
```

This Example shows a Heart on the LED Matrix on the Arduino UNO R4 WiFi



LED Matrix Editor




```
const uint32_t heart[] =
{
  0x3184a444,
  0x44042081,
  0x100a0040
};
```

https://docs.arduino.cc/tutorials/uno-r4-wifi/led-matrix#animation-generation

```
#include "Arduino_LED_Matrix.h"

const uint32_t heart[] = {
    0x3184a444,
    0x44042081,
    0x100a0040
};

ArduinoLEDMatrix matrix;

void setup()
{
  Serial.begin(115200);
  matrix.begin();
  matrix.loadFrame(heart);
}

void loop()
{

}
```

Code created by the LED Matrix Editor

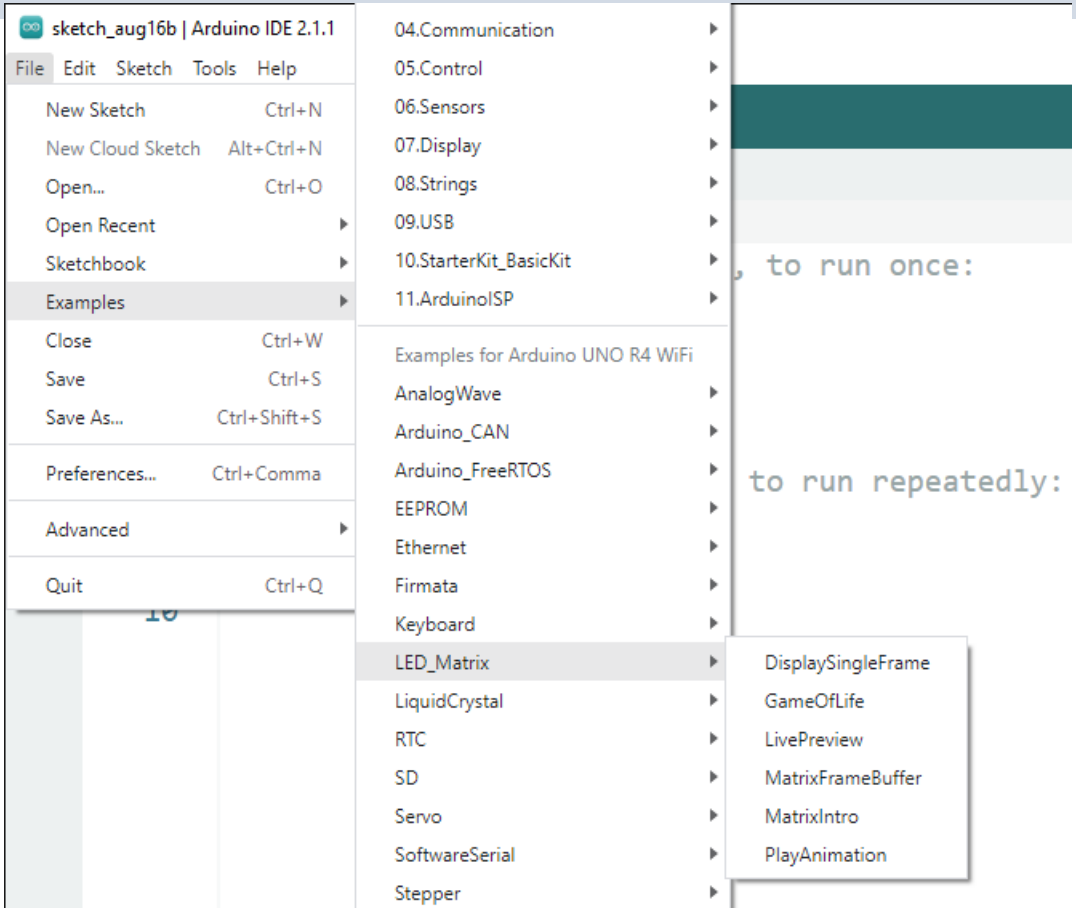
This Example shows a Heart on the LED Matrix on the Arduino UNO R4 WiFi

```
#include "Arduino_LED_Matrix.h"

const uint32_t heart[] = {
    0x3184a444,
    0x44042081,
    0x100a0040
};

const uint32_t happy[] = {
    0x19819,
    0x80000001,
    0x81f8000
};

ArduinoLEDMatrix matrix;

void setup()
{
  Serial.begin(115200);
  matrix.begin();

}

void loop()
{
  matrix.loadFrame(heart);
  delay(1000);
  matrix.loadFrame(happy);
  delay(1000);
}
```

Code created by the LED Matrix Editor

This Example switch between showing a Heart and a Happy face on the LED Matrix on the Arduino UNO R4 WiFi

More Examples



Arduino Pins

Hans-Petter Halvorsen

Table of Contents

Arduino Pins

- Digital
  - Digital Out
  - Digital In
  - PWM
- Analog Out (DAC)
- Analog In

We will provide some basic code examples where we test the different Pins

External LED

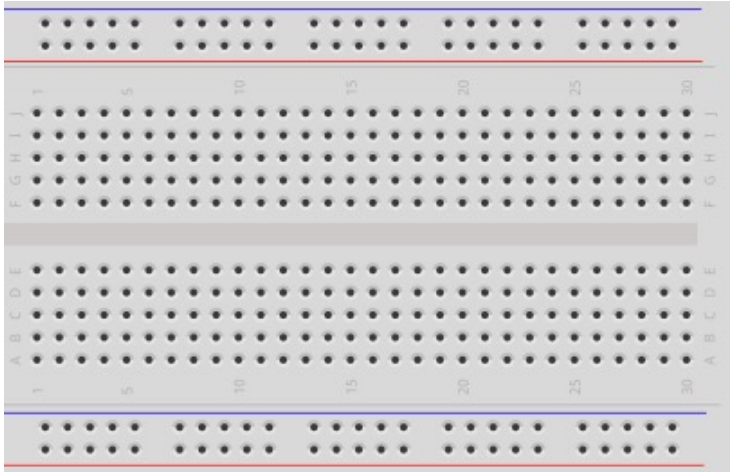
Hans-Petter Halvorsen

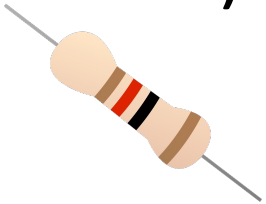
Table of Contents

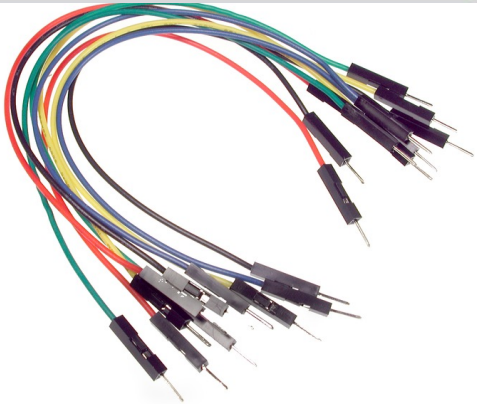
External LED Example

What do we need?

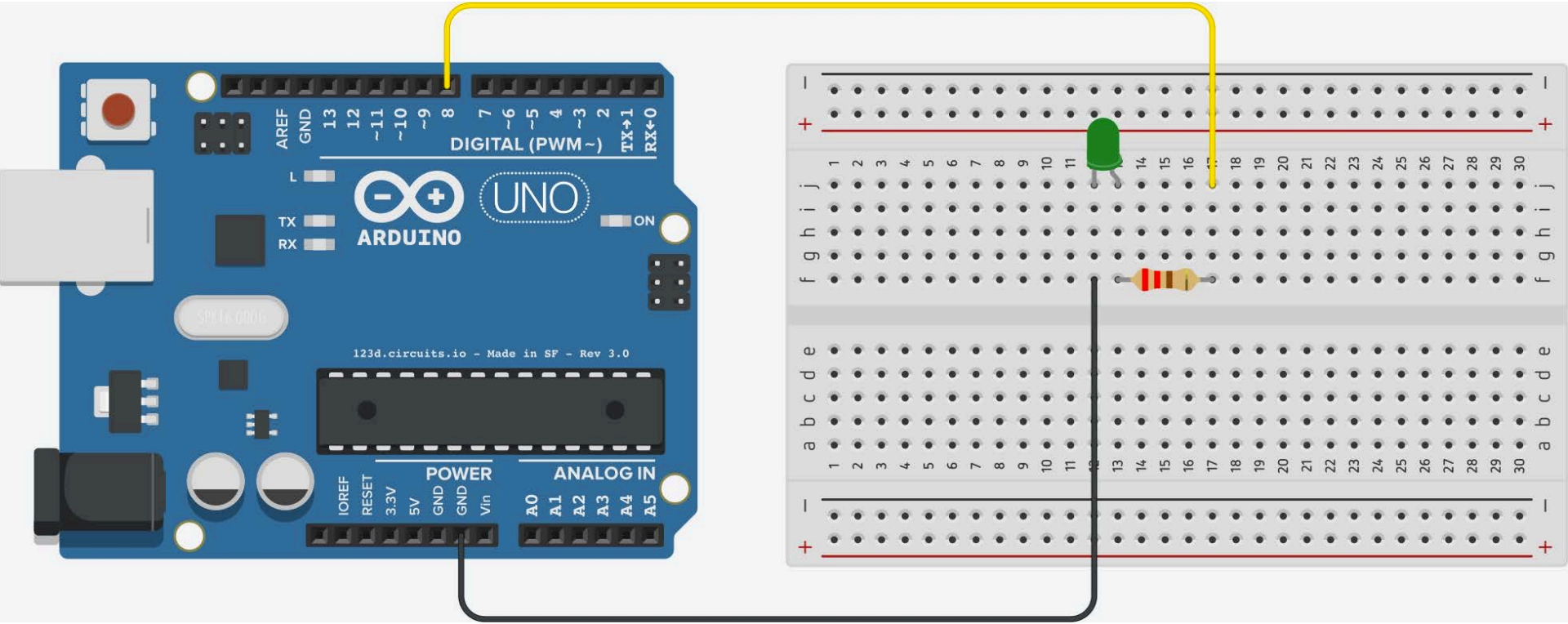
- Breadboard
- LED
- Wires
- Resistor (e.g., $R = 270\Omega$)







External LED Wiring



Blinking External LED

```
int ledPin = 8;
void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

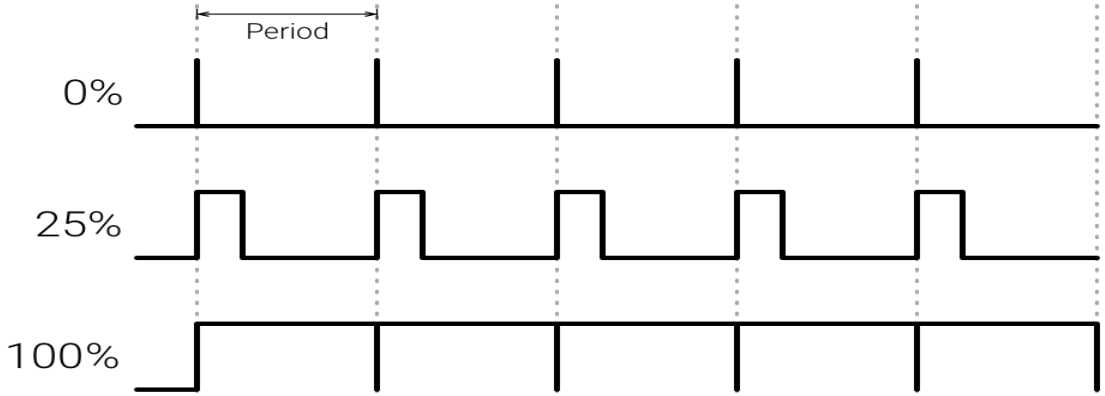
PWM

Pulse Width Modulation

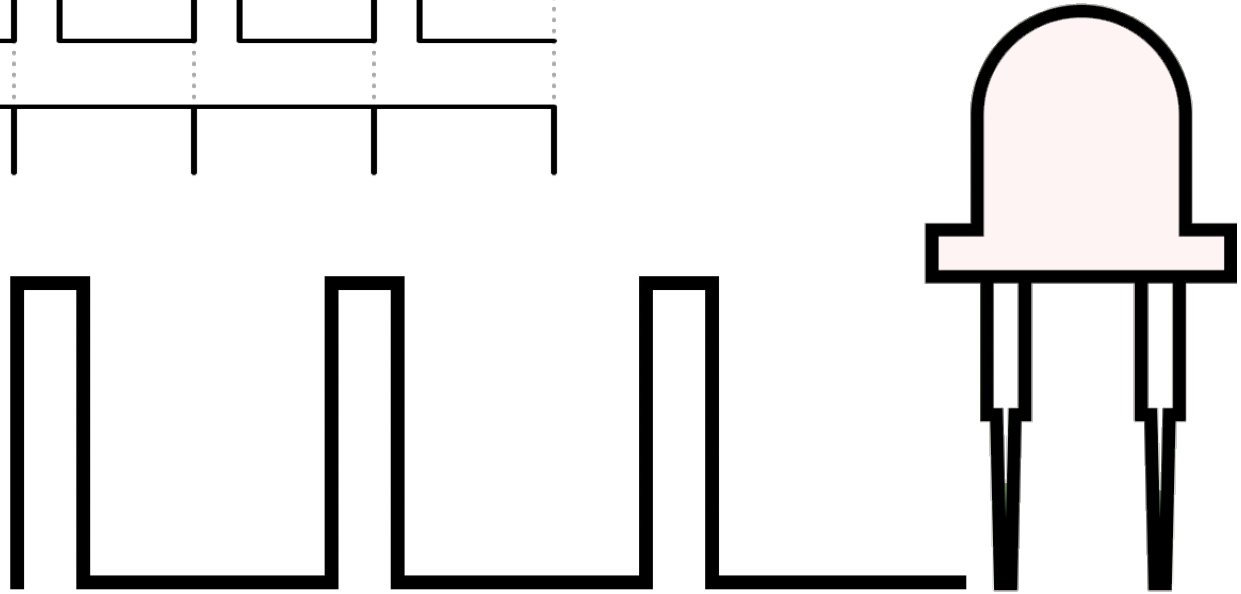
Hans-Petter Halvorsen

Table of Contents

Pulse Width Modulation (PWM)

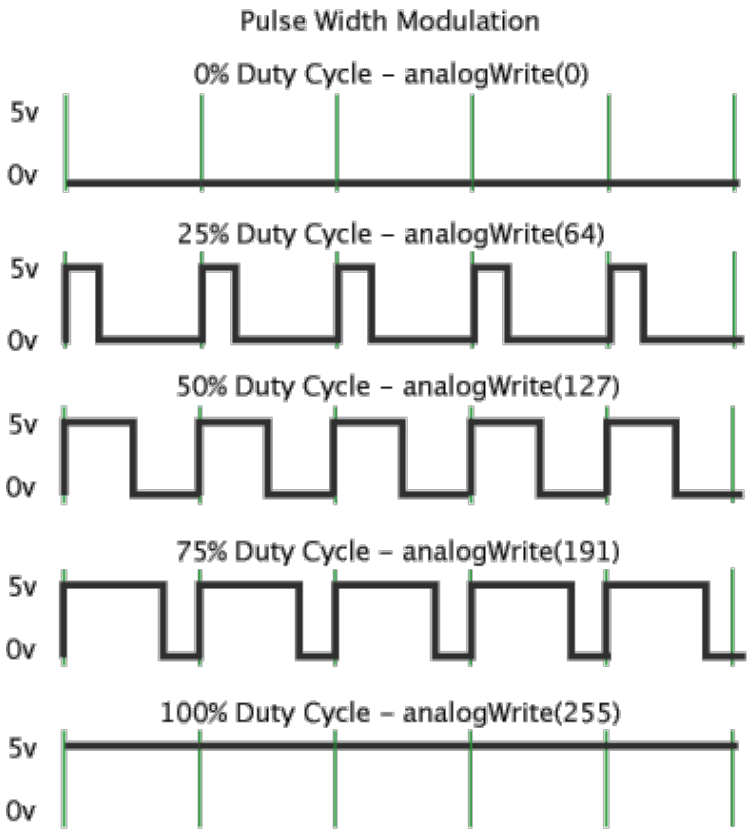


We can use PWM to control the brightness of a LED and many other things



https://www.electronicwings.com/raspberry-pi/raspberry-pi-pwm-generation-using-python-and-c

Pulse Width Modulation (PWM)



The Raspberry Pi UNO R4 Minima supports PWM on pins marked with ~
These pins are 3, 5, 6, 9, 10, 11

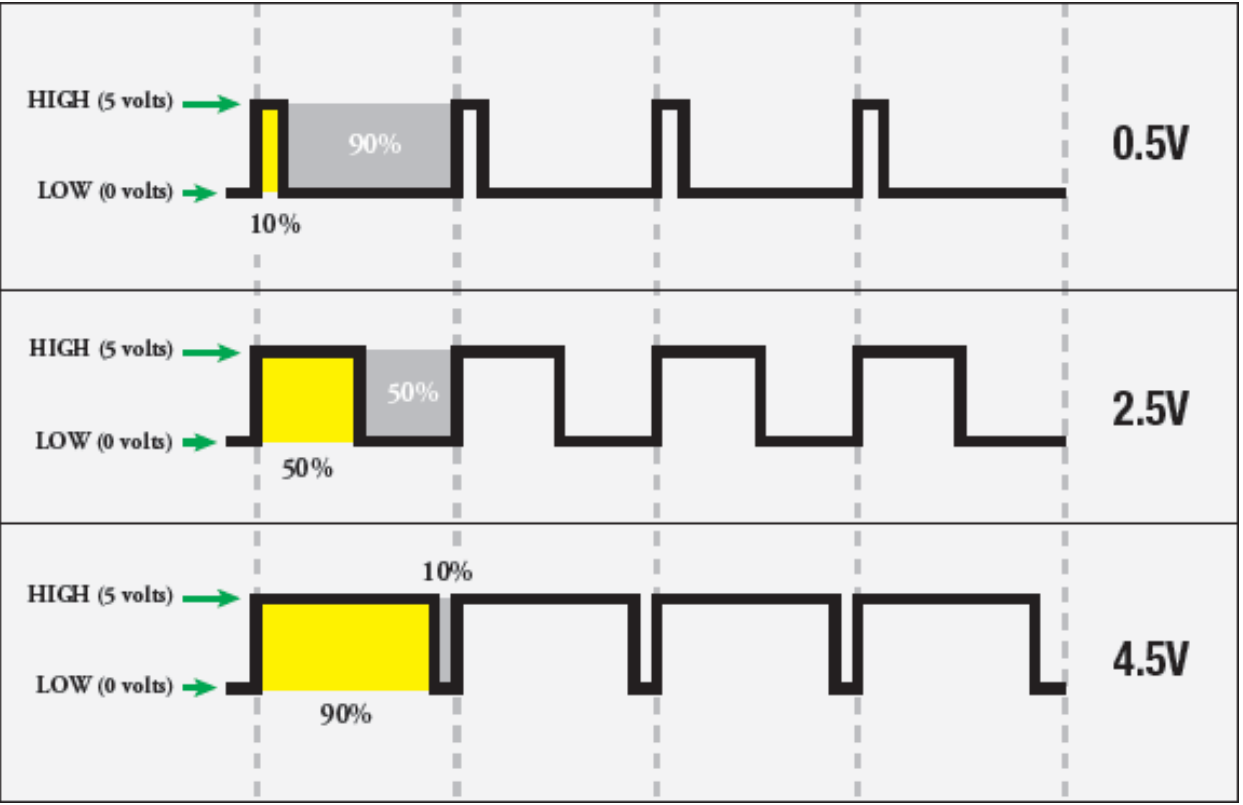
We use the following Arduino Function for PWM:

```
analogWrite(pin, value);
```

Note that despite the function name, the output is a digital signal, often referred to as a square wave

https://docs.arduino.cc/learn/microcontrollers/analog-output

Pulse Width Modulation (PWM)



The average voltage Level is 0.5V

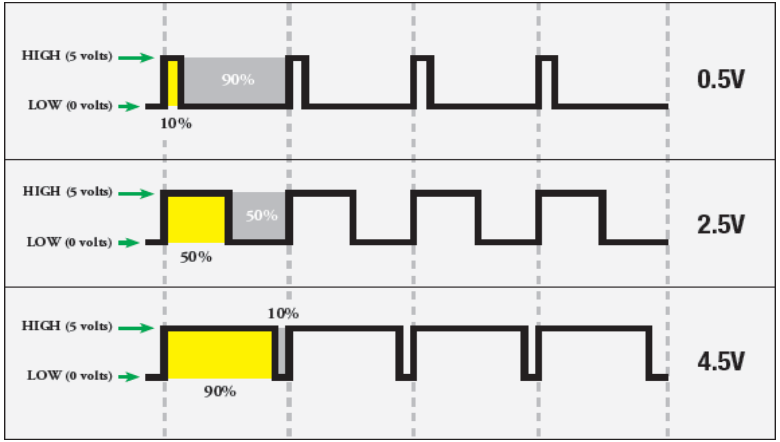
The average voltage Level is 2.5V

The average voltage Level is 4.5V

https://learn.sparkfun.com/tutorials/pulse-width-modulation/

analogWrite

Arduino can give a signal between 0 and $5V$



$$0.5V: \frac{0.5}{5} 100\% \rightarrow 10\%$$

0.5V (10% of 255) -> analogWrite(pin, 25)

$$2.5V: \frac{2.5}{5} 100\% \rightarrow 50\%$$

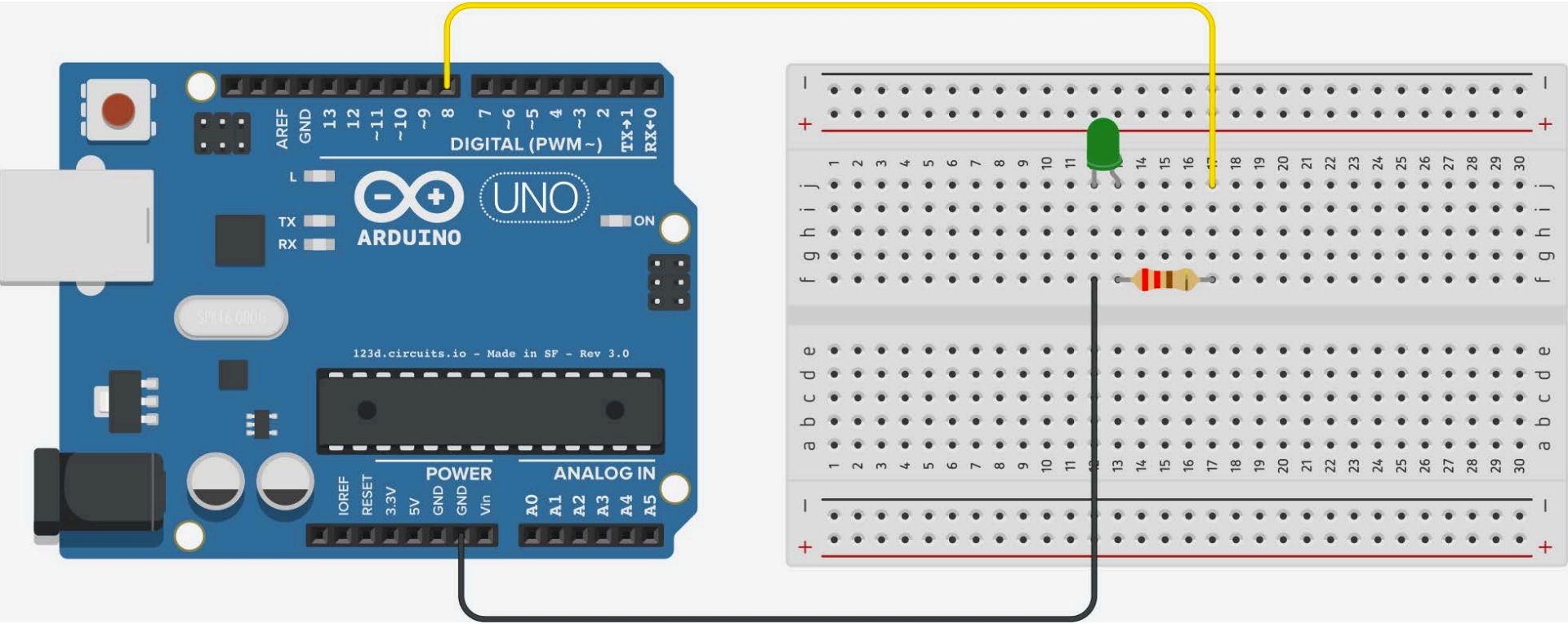
2.5V (50% of 255) -> analogWrite(pin, 127)

$$4.5V: \frac{4.5}{5} 100\% \rightarrow 90\%$$

4.5V (90% of 255) -> analogWrite(pin, 229)

Arduino syntax: **analogWrite(pin, value)**
value: the duty cycle: between 0 (always off) and 255 (always on).
0-5V -> 0-255

Wiring


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
123d.circuits.io - Made in SF - Rev 3.0
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

PWM Example

255 -> Max Brightness

0 -> Min Brightness -> LED Off

```
int ledPin = 3;
int value;

void setup()
{
   pinMode(ledPin, OUTPUT);
}

void loop()
{
  value = 255;
  analogWrite(ledPin, value);
  delay(1000);

  value = 0;
  analogWrite(ledPin, value);
  delay(1000);
}
```

PWM Example – random()

```
int ledPin = 3;
int value = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  value = random(256);
  analogWrite(ledPin, value);
  delay(1000);
}
```

PMM - Voltage

Typically, we want to specify the Voltage value

0-5V -> 0-255

0V -> 0

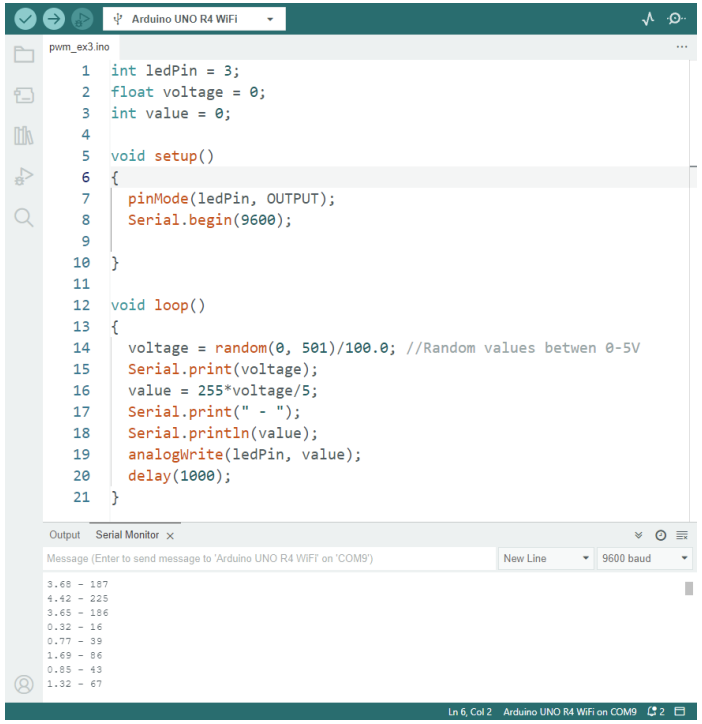
1V ->

5V -> 255

$$y = \frac{255}{5}x$$

x: 0-5V

y: 0-255




```
int ledPin = 3;
float voltage = 0;
int value = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);

}

void loop()
{
  voltage = random(0, 501)/100.0;
  Serial.print(voltage);
  value = 255*voltage/5;
  Serial.print(" - ");
  Serial.println(value);
  analogWrite(ledPin, value);
  delay(1000);
}
```

PMM - Voltage

0-5V -> 0-255

0V -> 0
1V ->
..
5V -> 255
This gives this formula that w need to use:

$$y = \frac{255}{5}x$$

x: 0-5V
y: 0-255

Arduino UNO R4 WiFi

pwm_ex3.ino

```
int ledPin = 3;
float voltage = 0;
int value = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);

}

void loop()
{
  voltage = random(0, 501)/100.0; //Random values betwen 0-5V
  Serial.print(voltage);
  value = 255*voltage/5;
  Serial.print(" - ");
  Serial.println(value);
  analogWrite(ledPin, value);
  delay(1000);
}
```

Output | Serial Monitor

Message (Enter to send message to 'Arduino UNO R4 WiFi' on 'COM9') | New Line | 9600 baud

```
3.68 - 187
4.42 - 225
3.65 - 186
0.32 - 16
0.77 - 39
1.69 - 86
0.85 - 43
1.32 - 67
```

Ln 6, Col 2 Arduino UNO R4 WiFi on COM9 2

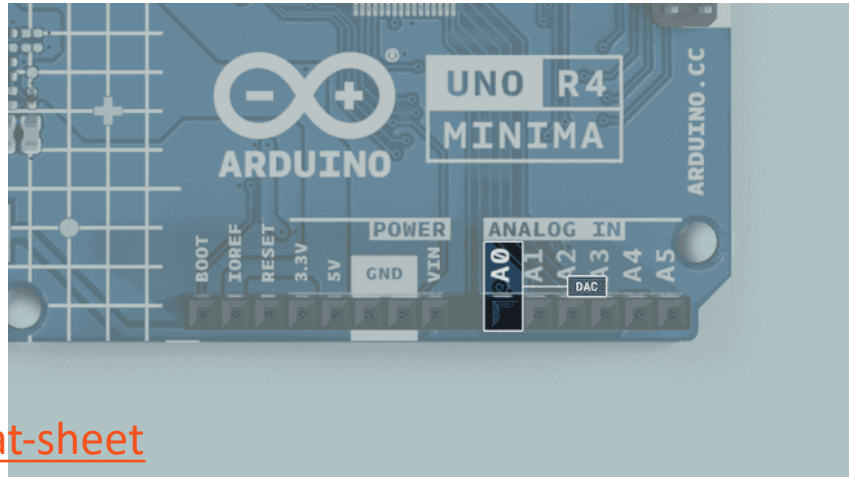
Analog Out (DAC)

Hans-Petter Halvorsen

Table of Contents

Analog Out

- Arduino UNO R3 has no real Analog Out pins, only PWM
- The UNO R4 has a DAC with up to 12-bit resolution



https://docs.arduino.cc/tutorials/uno-r4-minima/cheat-sheet

Analog Out

We use the analogWrite() function in order to use the DAC:

```
analogWrite(pin, value);
```

```
int analogPin = A0;
int value = 255;
analogWrite(analogPin, value);
```

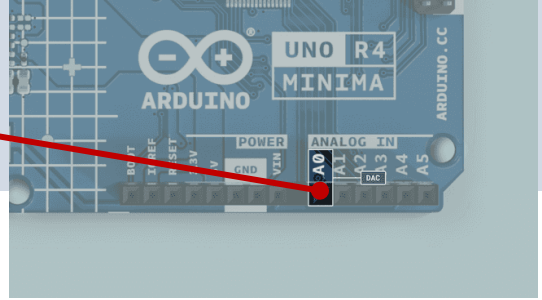
This DAC pin has a default write resolution of 8 bits. This means that values that are written to the pin should be between 0-255 (2^8).

```
analogWriteResolution(12);
```

You may change this resolution to up to 12 bits. The values you write to the pin should be between 0-4096 (2^12)

https://docs.arduino.cc/tutorials/uno-r4-minima/cheat-sheet

```
int analogPin = A0;

void setup()
{
  analogWriteResolution(8);
}

void loop()
{
  int value = 255;
  analogWrite(analogPin, value);
  delay(1000);

  value = 0;
  analogWrite(analogPin, value);
  delay(1000);
}
```



```
int analogPin = A0;
float voltvalue;
int dacvalue;

void setup()
{
  analogWriteResolution(8);
}

void loop()
{
  voltvalue = 5;
  dacvalue = VoltToDac(voltvalue);
  analogWrite(analogPin, dacvalue);
  delay(1000);

  voltvalue = 0;
  dacvalue = VoltToDac(voltvalue);
  analogWrite(analogPin, dacvalue);
  delay(1000);
}

int VoltToDac(float volt)
{
  int dacmax = 255;
  int dac = dacmax*volt/5;
  return dac;
}
```

Typically, we want to deal with voltage values instead of DAC values

A Function is made to convert from Voltage (0-5V) to DAC Value (0-255)

Analog In (ADC)

Hans-Petter Halvorsen

Table of Contents

Analog In/ADC

- Arduino UNO R4 has 6 Analog In pins
  - Use the analogRead(pin) Function
- An analog-to-digital converter (ADC) transforms an analog signal to a digital one
- By default, the ADC resolution is set to 10 bit (2^10=1024), i.e., values between 0 and 1023
- It can be updated to 12 bit (2^12=4096) or 14 bit (2^14=16384) if you need higher accuracy
  - Use the analogReadResolution(bit) Function

https://docs.arduino.cc/tutorials/uno-r4-minima/adc-resolution

Analog In

```
int analogReadPin = A1;
int valueAdc = 0;
float valueVoltage = 0;
void setup()
{
  analogReadResolution(10);
  Serial.begin(9600);
}

void loop()
{
  valueAdc = analogRead(analogReadPin);
  valueVoltage = (5*valueAdc)/1023.0;
  Serial.println(valueVoltage);
  delay(5000);
}
```

File Edit Sketch Tools Help

Arduino UNO R4 WiFi

analog_in_ex.ino

```
1  int analogReadPin = A1;
2  int valueAdc = 0;
3  float valueVoltage = 0;
4  void setup()
5  {
6    analogReadResolution(10);
7    Serial.begin(9600);
8  }
9
10 void loop()
11 {
12   valueAdc = analogRead(analogReadPin);
13   valueVoltage = (5*valueAdc)/1023.0;
14   Serial.println(valueVoltage);
15   delay(5000);
16 }
```

Output Serial Mon

Message (Enter to s

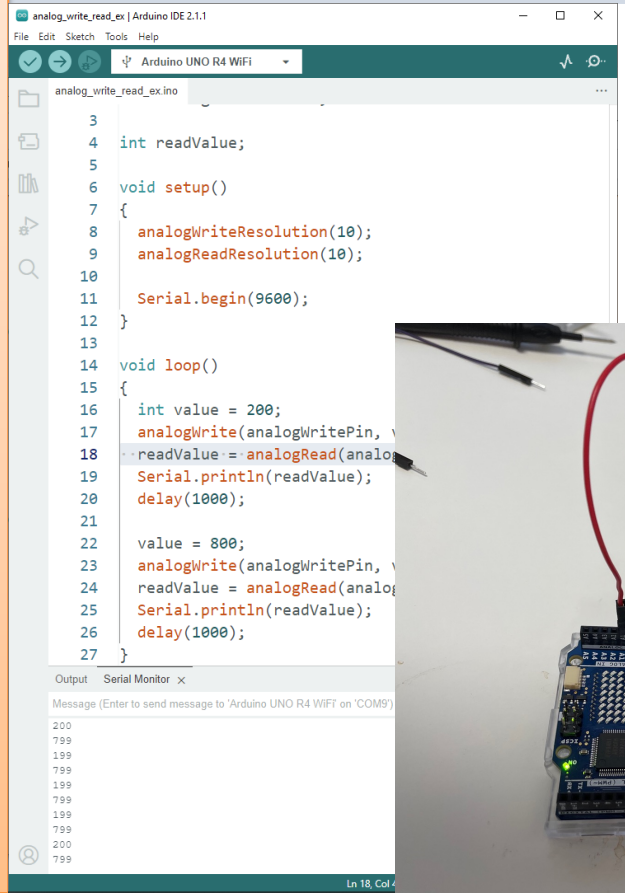
```
0.95
1.61
0.72
0.96
1.55
0.50
0.88
1.77
1.27
```

Here we have connected a 1.5V Battery to one of the Analog Input pins on Arduino. We convert the ADC Value (0-1023) to a Voltage Value (0-5V) and Print the Voltage Value to the Serial Monitor

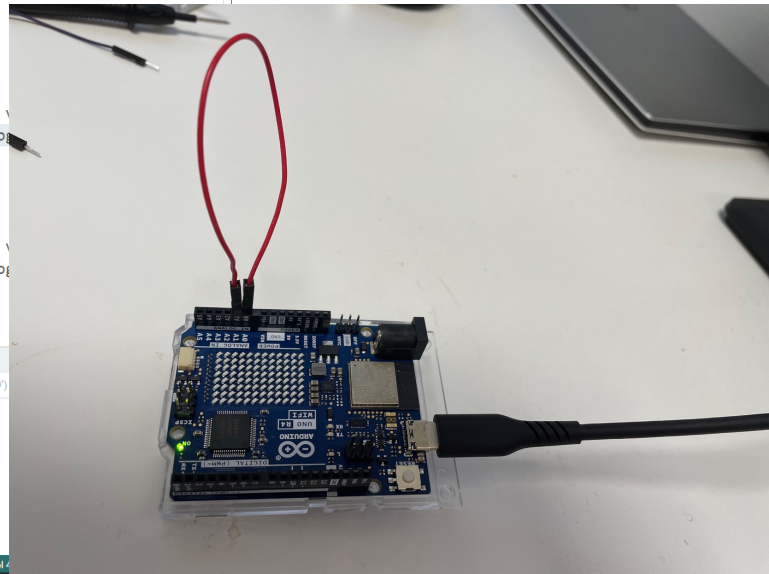
Ln 13, Col 29 Arduino UNO R4 WiFi on COM9 2

Analog Write and Read Example

```
int analogWritePin = A0;
int analogReadPin = A1;

int readValue;

void setup()
{
  analogWriteResolution(10);
  analogReadResolution(10);

  Serial.begin(9600);
}

void loop()
{
  int value = 200;
  analogWrite(analogWritePin, value);
  readValue = analogRead(analogReadPin);
  Serial.println(readValue);
  delay(1000);

  value = 800;
  analogWrite(analogWritePin, value);
  readValue = analogRead(analogReadPin);
  Serial.println(readValue);
  delay(1000);
}
```



In this example we have wired A0 and A1 together



TMP36

Analog In (ADC)

Hans-Petter Halvorsen

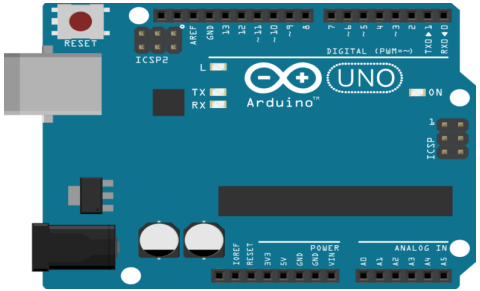
Table of Contents

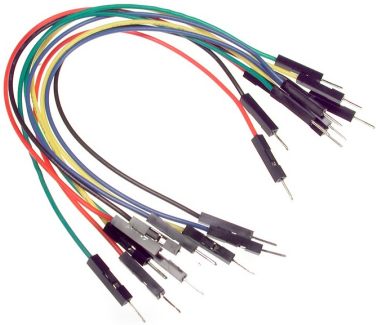
Temperature Sensor Example

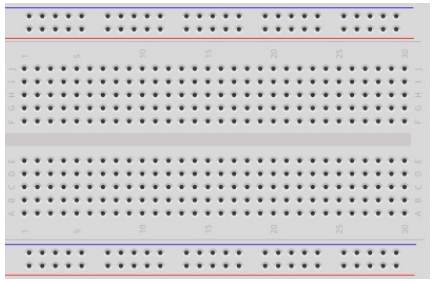
- In this example we will use a small temperature sensor to read the temperature in the room.
- The Temperature Sensor is called “TMP36”
- In this example we will use one of the "Analog In" ports on the Arduino board

Necessary Equipment

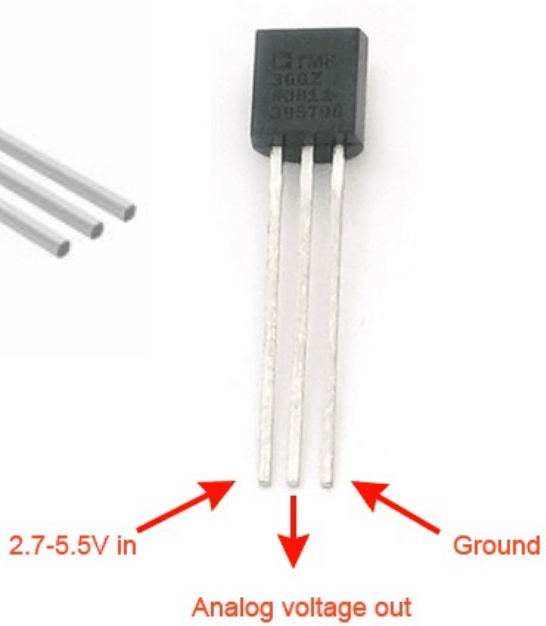
- Arduino
- Breadboard
- TMP36
- Wires (Jumper Wires)

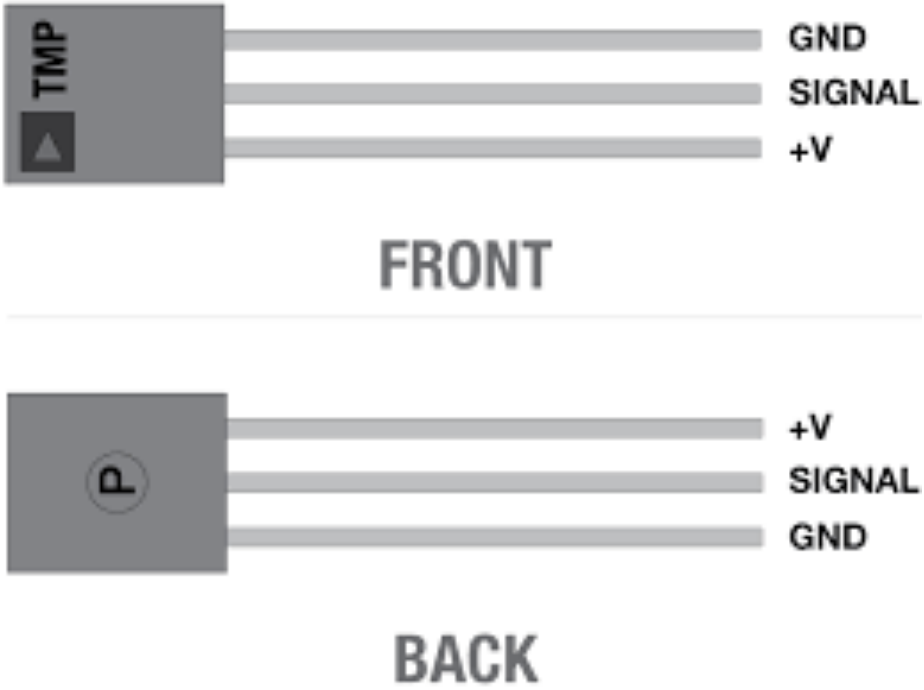






TMP36

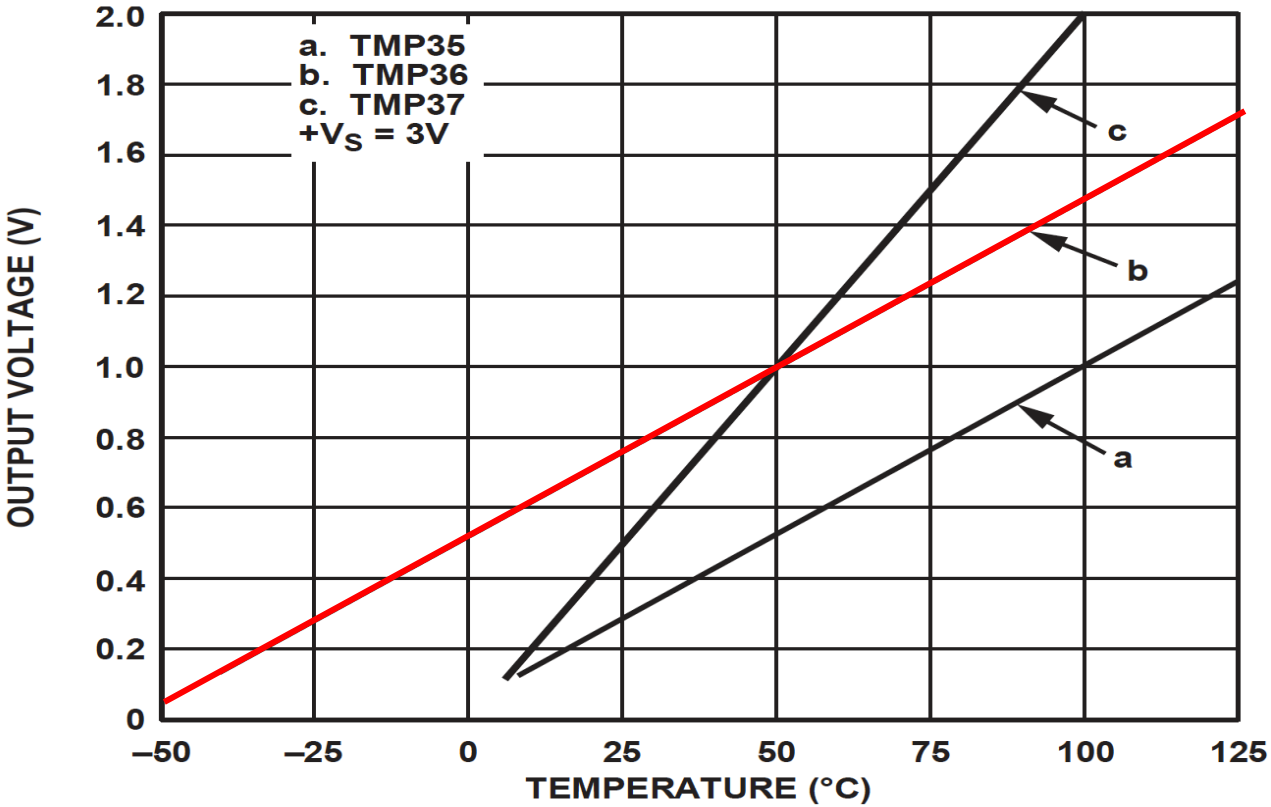




TMP is a small, low-cost temperature sensor and cost about $1 (you can buy it “everywhere”)

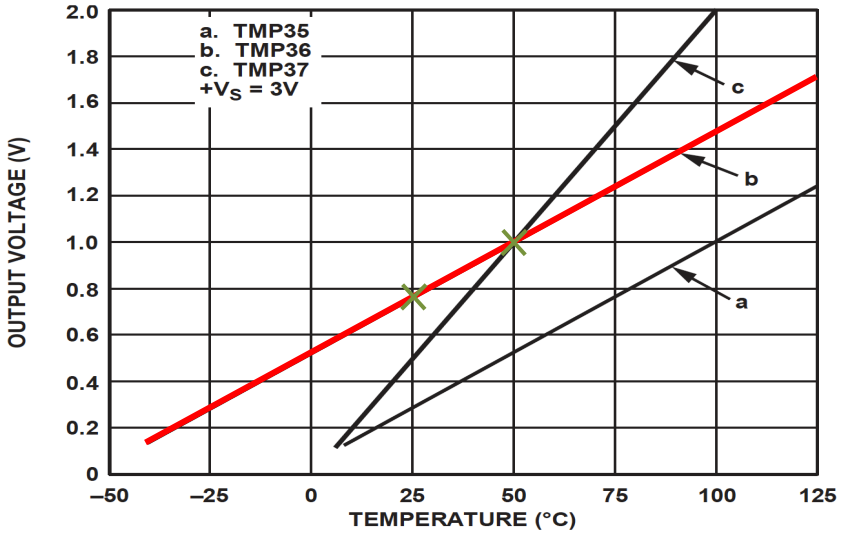
Datasheet

Output Voltage vs. Temperature



https://www.arduino.cc/en/uploads/Main/TemperatureSensor.pdf

Linear Scaling



Convert form Voltage (V) to degrees Celsius
From the Datasheet we have:

$$(x_1, y_1) = (0.75V, 25°C)$$
$$(x_2, y_2) = (1V, 50°C)$$

There is a linear relationship between Voltage and degrees Celsius:

$$y = ax + b$$

We can find a and b using the following known formula:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

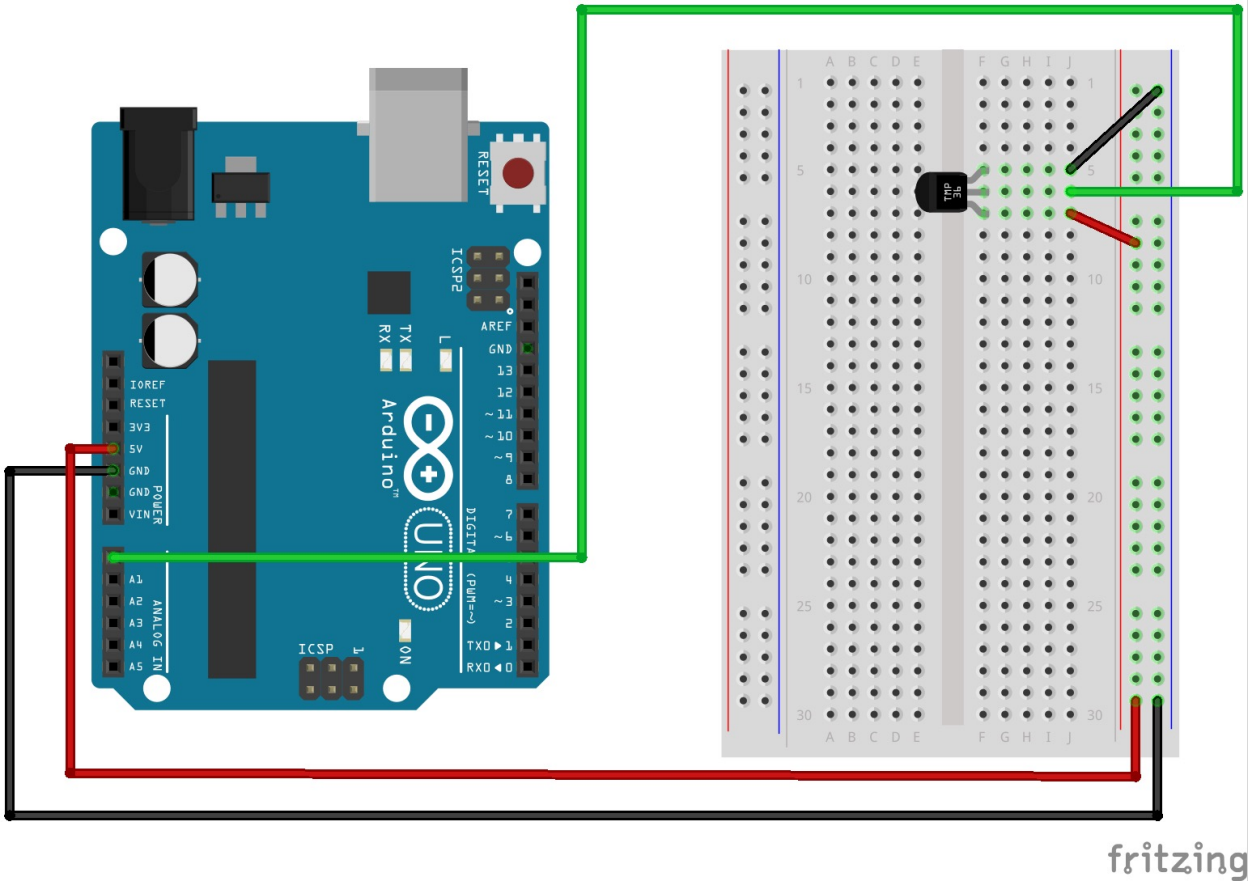
This gives:

$$y - 25 = \frac{50 - 25}{1 - 0.75}(x - 0.75)$$

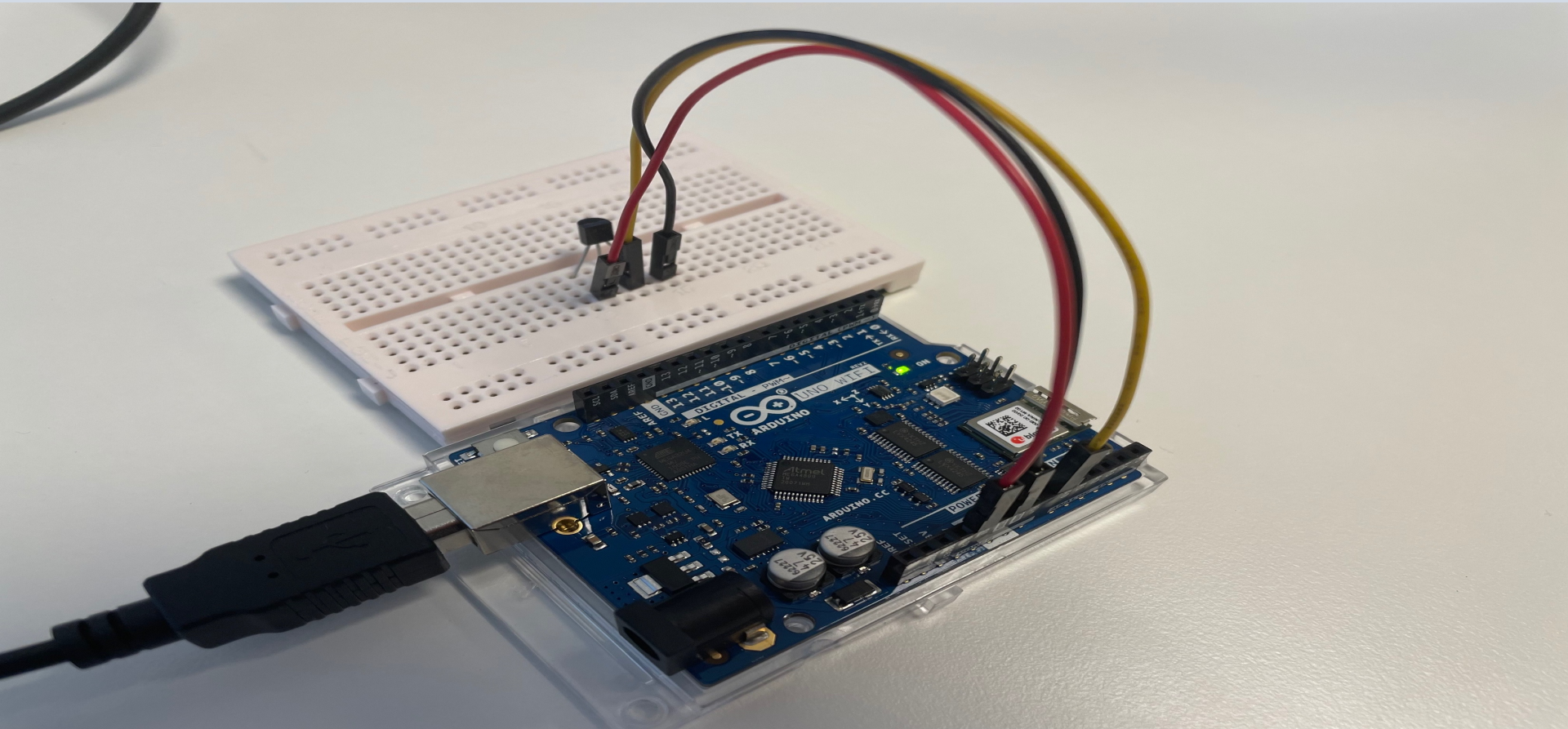
Then we get the following formula:

$$y = 100x - 50$$

Wiring


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino
UNO
ON
ICSP
1
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A1
A2
A3
A4
A5
ANALOG IN
A B C D E
F G H I J
1
5
10
15
20
25
30
fritzing

Wiring



Temperature Conversion

We want to present the value from the sensor in degrees Celsius:

1. The function analogRead() gives a value between 0 and 1023 (Arduino UNO has a built-in 10-bit ADC, 2^10=1024)
2. Then we convert this value to 0-5V.
3. Finally, we convert to degrees Celsius using information from the Datasheet presented on the previous page ($y = 100x - 50$)
4. The we can, e.g., show the Temperature value in the Serial Monitor

Code

```
const int temperaturePin = 0;

float adcValue;
float voltage;
float degreesC;

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

void loop()
{

  adcValue = analogRead(temperaturePin);

  voltage = (adcValue*5)/1023;

  degreesC = 100*voltage - 50;

  Serial.print("ADC Value: ");
  Serial.print(adcValue);

  Serial.print("  voltage: ");
  Serial.print(voltage);

  Serial.print("  deg C: ");
  Serial.println(degreesC);

  delay(1000);
}
```

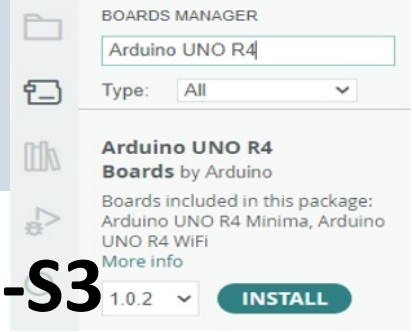
WiFi

Connection Arduino UNO R4 WiFi to Internet

Hans-Petter Halvorsen

Table of Contents

WiFi



- **Arduino UNO R4 WiFi** has a built in **ESP32-S3** module that enables you to connect to Wi-Fi networks and perform network operations.
- Wi-Fi support is enabled via the built-in **WiFiS3 library** that is shipped with the Arduino UNO R4 Core.
- Installing the Arduino UNO R4 Core automatically installs the WiFiS3 library.

https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples

```
#include <WiFiS3.h>
#include "secrets.h"

char ssid[] = SECRET_SSID;
char pass[] = SECRET_PASS;
int status = WL_IDLE_STATUS;

void setup()
{
  Serial.begin(9600);
  ConnectWiFi();
}

void loop() {
  delay(10000);
  PrintNetwork();
}
```

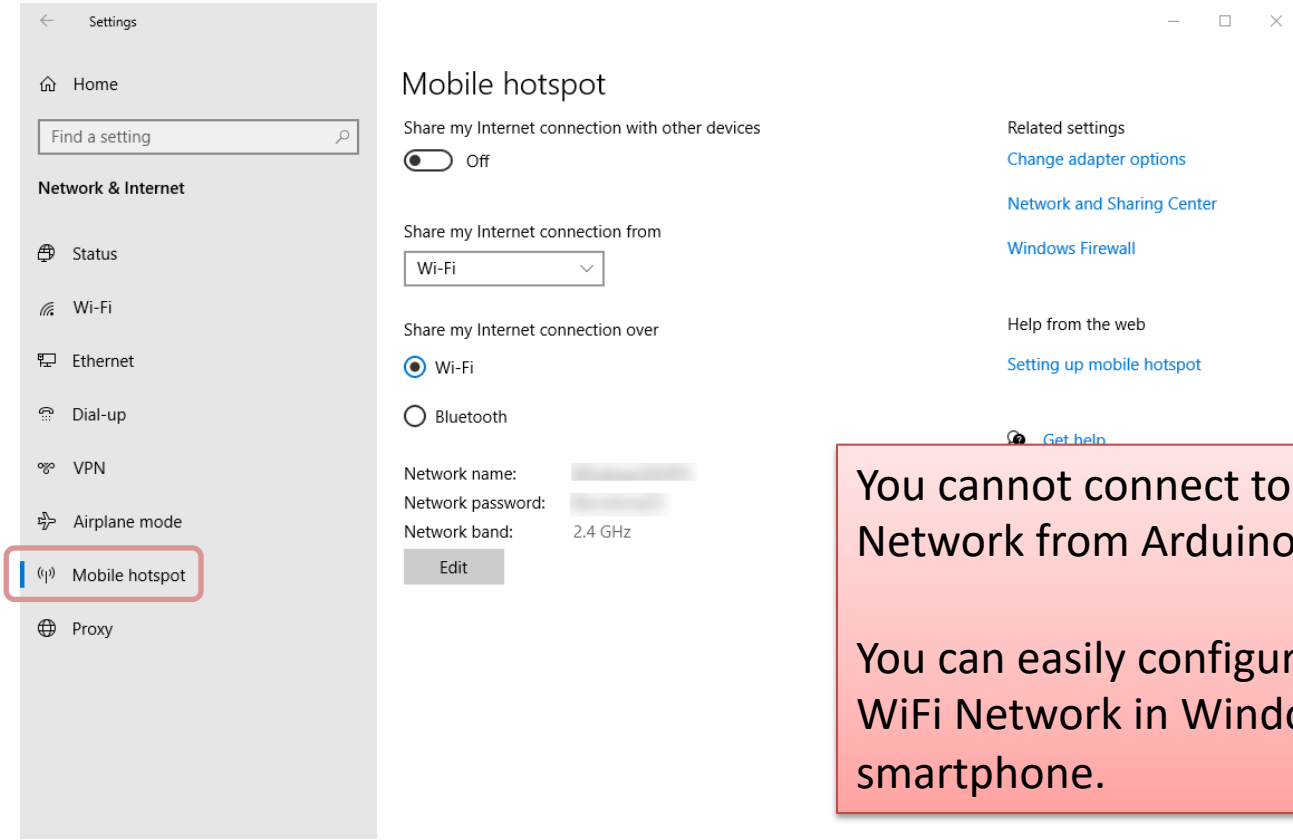
```
void PrintNetwork()
{
  Serial.print("WiFi Status: ");
  Serial.println(WiFi.status());

  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```

```
void ConnectWiFi()
{
  // check for the WiFi module:
  if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    while (true);
  }

  String fv = WiFi.firmwareVersion();
  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
  }

  // Attempt to connect to WiFi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, pass);

    // wait 10 seconds for connection:
    delay(10000);
  }

  // You're connected now, so print out the data:
  Serial.println("You're connected to Wifi");
  PrintNetwork();
}
```

“secrets.h”:

```
#define SECRET_SSID "xxx"
#define SECRET_PASS "xxx"
```

Setting up a Mobile Hotspot WiFi Network



You cannot connect to your Eduroam Network from Arduino.

You can easily configure a Mobile Hotspot WiFi Network in Windows 10/11 or on your smartphone.

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog