Arduino

The basics

Pierre Rossel
2021

Setup and loop

- An arduino program is composed of at least 2 functions
  - Setup
    - Executed only once at the beginning of the program
  - Loop
    - Executed in a loop

Output

- pinMode(13, OUTPUT)
  - Defines that pin 13 is to be used as an output
- digitalWrite(13, HIGH)
  - Sends 5V to pin 13
- digitalWrite(13, LOW)
  - Sends 0V to pin 13

Example: Turn on the LED

```
// Turns on the LED on pin 13
void setup() {
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
}

void loop() {
}
```

Loop

- Function loop() is called in a loop
- Example

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Constant

- Allows you to give a name and a meaning to a value
- The value cannot change during execution
- Allows you to isolate the numerical values (the configuration) in one place

```
#define LED 13

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

Variable

- Allows you to assign a name to a value
- The value can change during the execution of the program
- The value is used instead of the name
- Example

```
#define LED 13
boolean val = HIGH;
void setup() {
  pinMode(LED, OUTPUT);
  digitalWrite(LED, val);
}
```

Input

- Read a digital value

```
#define BUTTON 8
pinMode(BUTTON, INPUT);
boolean val = digitalRead(BUTTON);
```

- Possible states
  - LOW (0V)
  - HIGH (5V)

Input

- Example

```
#define LED 13
#define BUTTON 8

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}
void loop() {
  boolean val = digitalRead(BUTTON);
  digitalWrite(LED, val);
}
```

Input

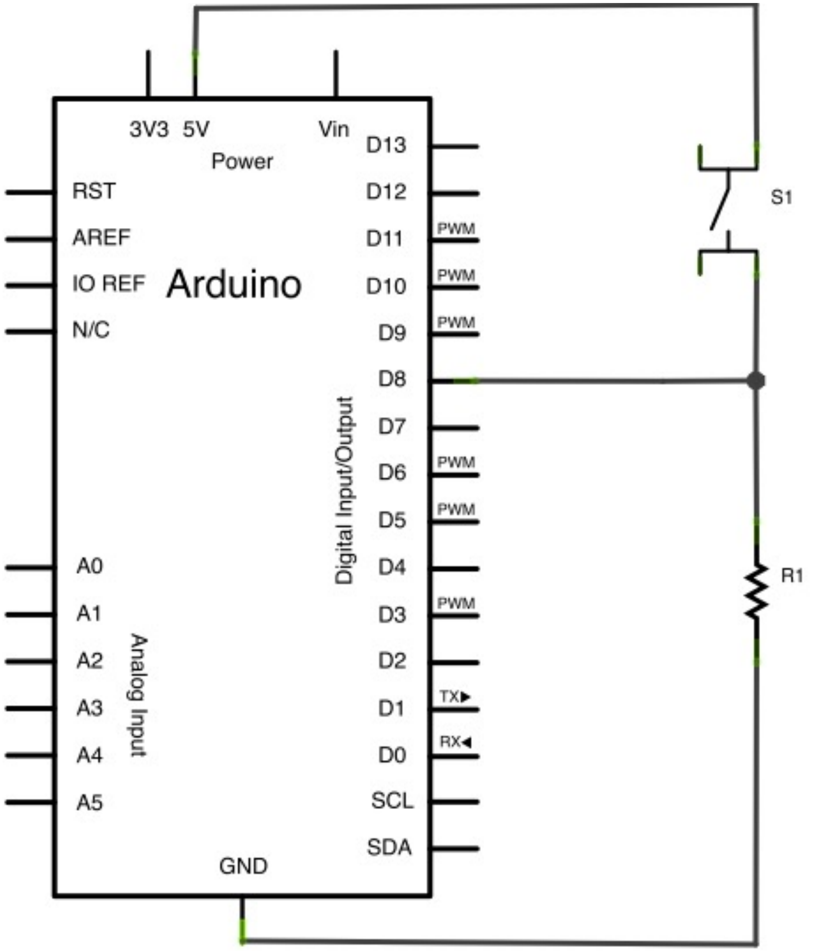
- Connect a wire between digital input 8 and 5V
  - the LED lights up
- Disconnect the 5V wire and put it on GND
  - The LED goes out
- If the wire is not connected, the state of the LED is not defined and can even change by itself

Pull-down

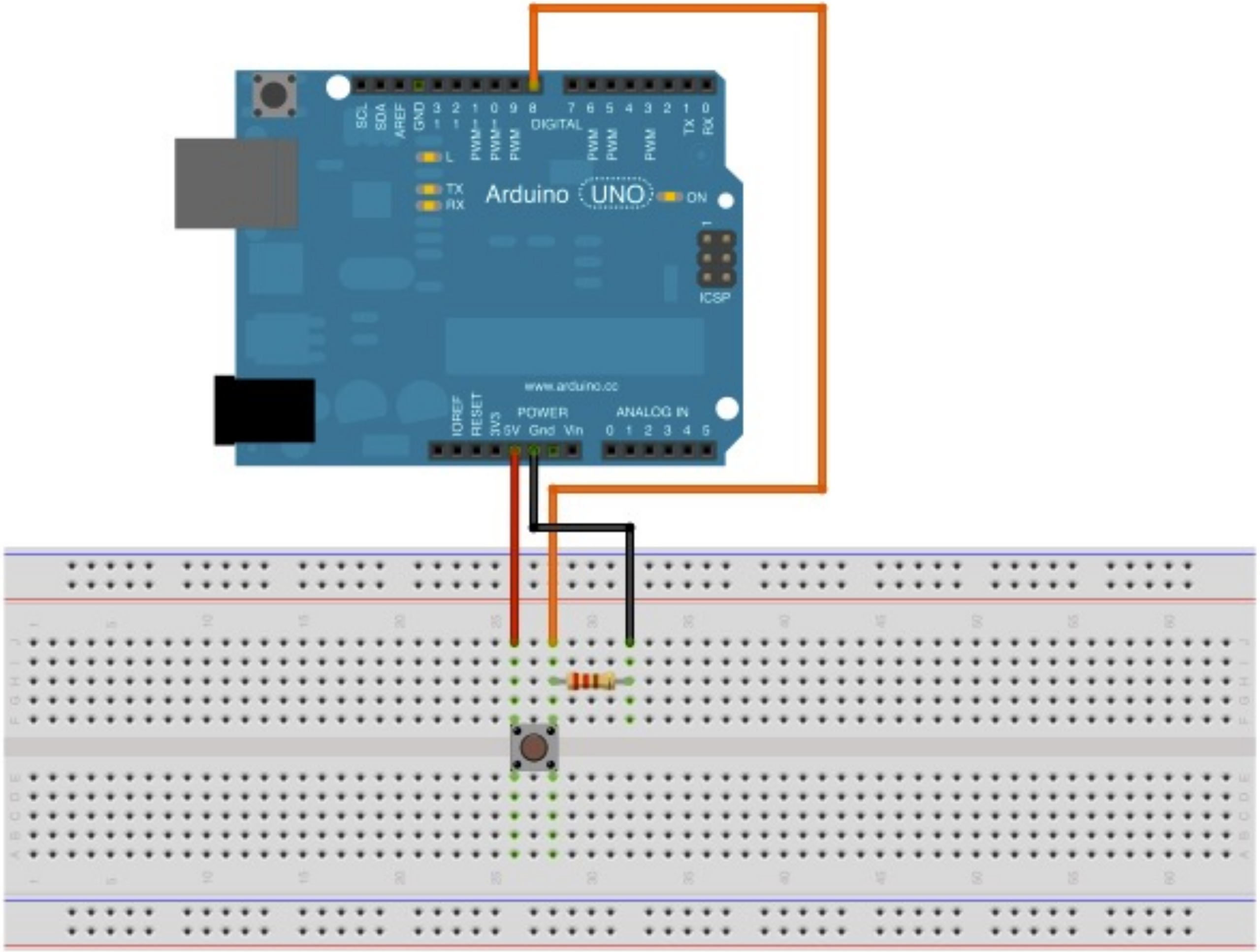
- A 1MOhm resistor between input 8 and GND allows to pull down the input to 0V
- If we want to switch input 8 to HIGH, we connect it to 5V with the wire
  - A small current passes through the resistor, but the input is well at 5V
- The wire can be replaced by a button
  - When pressed, the input is HIGH, otherwise it is LOW

Pull-down

- R1 pulls D8 down (0V) when S1 is open
- When S1 is closed, D8 is directly connected to 5V

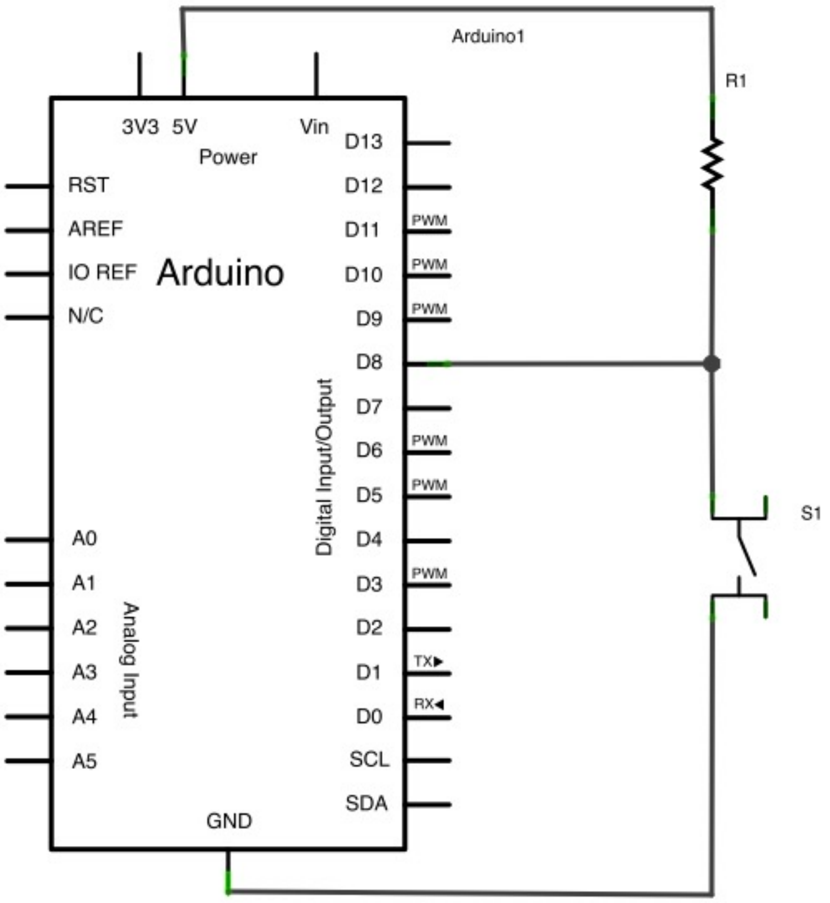


Pull-down


Arduino UNO
DIGITAL
ANALOG IN
POWER
ICSP
www.arduino.cc
5V
Gnd
Vin
TX
RX
ON
L

Pull-up

- Reverse logic
- The pull-up resistor pulls the input up (5V / HIGH) by default
- A button allows to connect the input to 0V (LOW)



Integrated pull-up

- Arduino has a built-in pull-up resistor (20KΩ)
- To activate it

```
pinMode(8, INPUT_PULLUP)
```

- No need for an external resistor
- The button must be connected to 0V
- Attention reversed logic
  - HIGH by default
  - LOW when the button is pressed

Serial communication

- Arduino can communicate with the computer by a serial link (via USB port)
  - In setup()

```
// initialize serial communication at 9600 bits per
  second:
Serial.begin(9600);
```

  - In loop()

```
boolean val = digitalRead(button);
Serial.println(val);
```

Reverse

- The "! " operator allows you to invert the value of a variable
- Example 1

```
digitalWrite(led, !val);
```

- Example 2

```
void setup() {
  Serial.begin(9600);
  boolean val = HIGH;
  Serial.println(val);
  val = !val;
  Serial.println(val);
}
```

Manual dimmer

```
#define LED 13
int lum = 15; // brightness between 0 and 20

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

void loop() {
  digitalWrite(LED, HIGH);
  delay(lum);
  digitalWrite(LED, LOW);
  delay(20 - lum);
}
```

Dimmer with button

```
// Read the status of the button
boolean val = digitalRead(BUTTON);

// Change the intensity
if (val) {
  lum = lum + 0.1;
  if (lum > 20) {
    lum = 0.0;
  }
}
```

SOS

- Send SOS signal in Morse code

```
// Ti
digitalWrite(LED, HIGH);
delay(200);
digitalWrite(LED, LOW);
delay(200);

// End of letter
delay(400);

// Your
digitalWrite(LED, HIGH);
delay(600);
digitalWrite(LED, LOW);
delay(200);
```

Functions

- Allows you to give a name to a code block
- Avoid duplication of identical code

```
  ti(); ti(); ti();

void ti(){
  digitalWrite(ledPin, HIGH);
  delay(iTiLength);
  digitalWrite(ledPin, LOW);
  delay(iTiLength);
}
```