ENGINE

Erasmus+

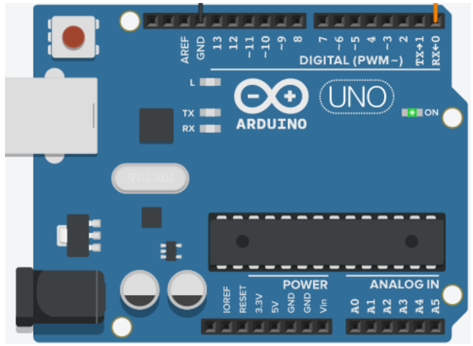
TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING
IN HIGHER EDUCATION

Module_1-2. Pins as inputs

Arduino Uno with Tinkercad

Contents

- Programming functions for the Arduino Uno
- Switches
- Push-buttons
- Example



Module_1-2. Pins as inputs

Programming functions

Functions that can be used in the Arduino Uno, as we saw in Module 1:

- pinMode(pin, value)
- digitalWrite(pin, value)
- delay(value).
- analogWrite(pin, value).

New function:

- digitalRead(pin): reads the state of pin and returns “0” or “1”

1. https://www.arduino.cc/reference/en/
2. https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/

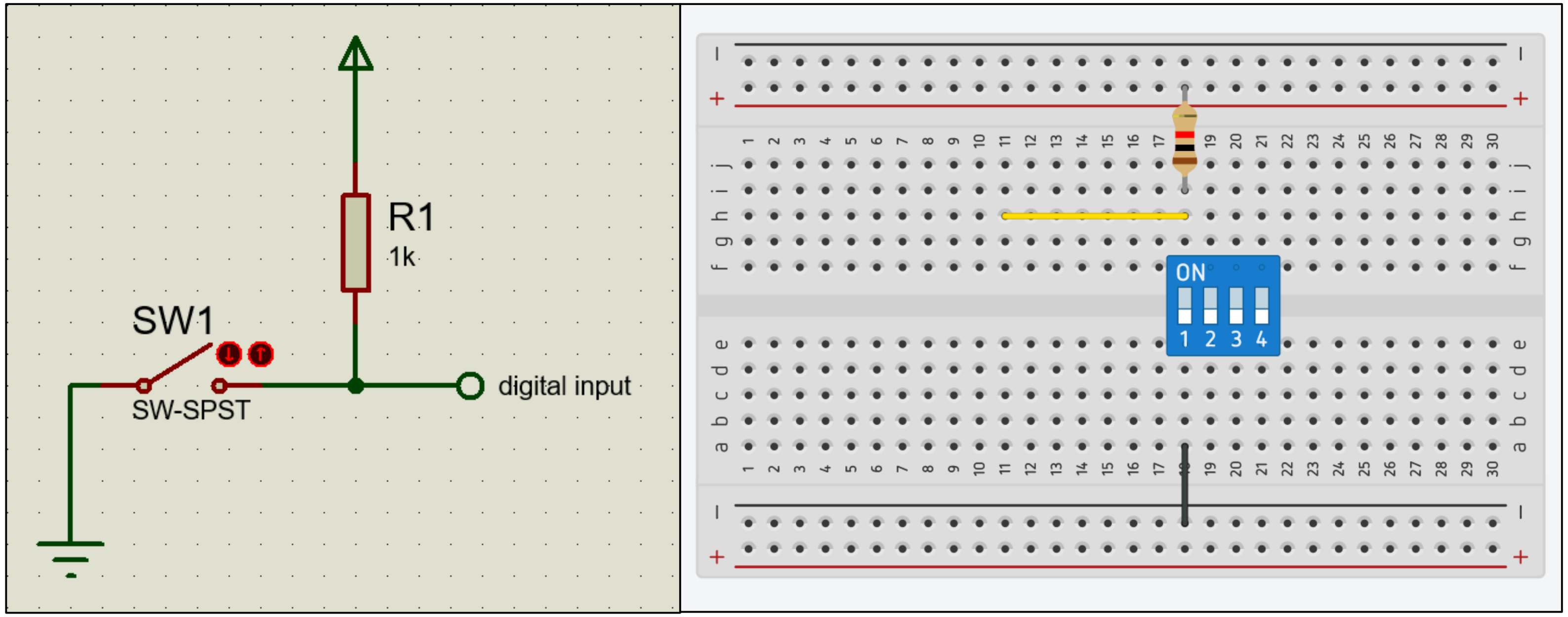
Module_1-2. Pins as inputs

Switches

Switches can be used in digital circuits to generate “0” and “1” using **pull-up** or pull-down resistors.

The digital input reads “1” while the switch is open.

When the switch is activated (closed) the input will read “0”.



Pull-up resistor

** The Arduino Uno has built-in pull-up resistors in its pins and can be activated by pinMode() **

1. https://www.seeedstudio.com/blog/2020/02/21/pull-up-resistor-vs-pull-down-differences-arduino-guide/

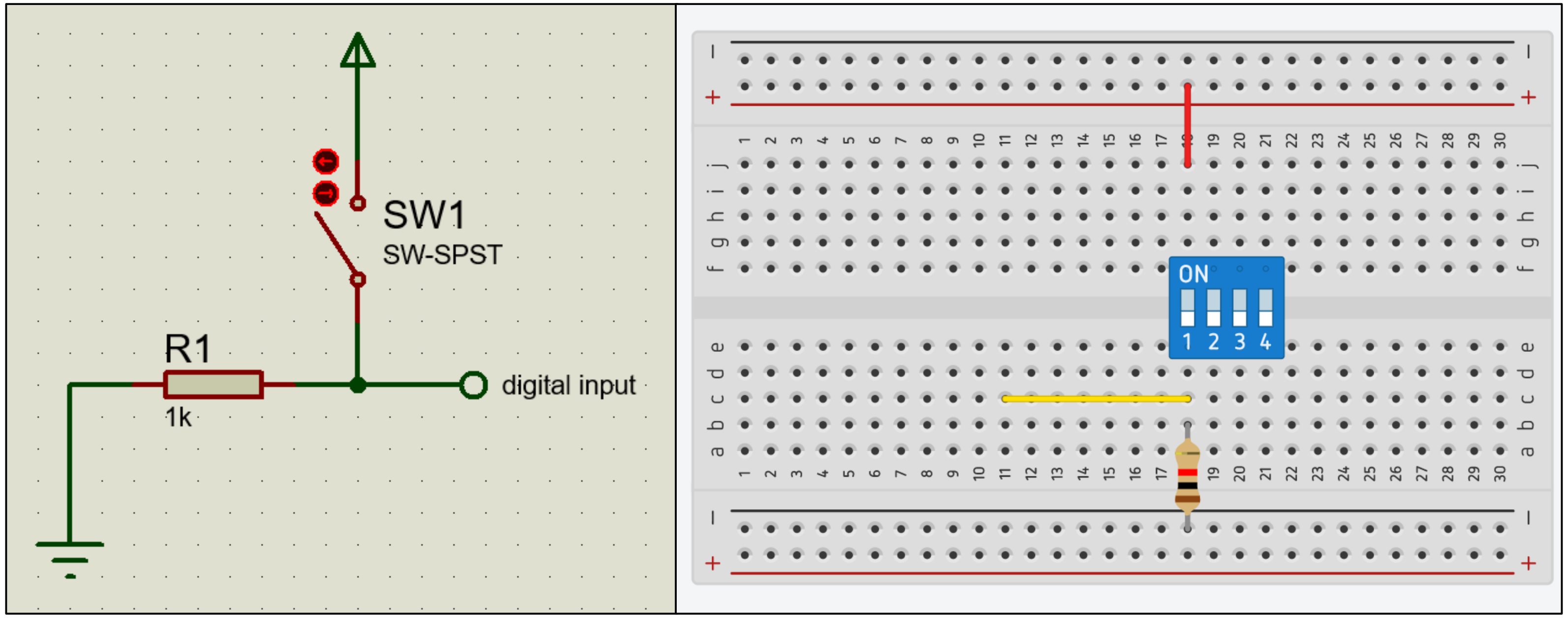
Module_1-2. Pins as inputs

Switches

Switches can be used in digital circuits to generate “0” and “1” using pull-up or **pull-down** resistors.

The digital input reads “0” while the switch is open.

When the switch is activated (closed) the input will read “1”.



Pull-down resistor

1. https://www.seeedstudio.com/blog/2020/02/21/pull-up-resistor-vs-pull-down-differences-arduino-guide/

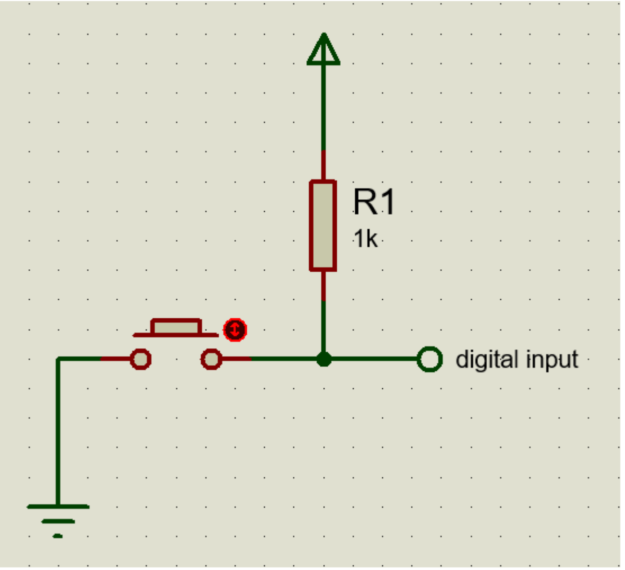
Module_1-2. Pins as inputs

Push-buttons

The push-button can be considered as a normally open switch, which closes for as long as we have pressed it. “0” and “1” can be created from the push-button with the appropriate wiring.

The digital input reads “1” as long as the push-button is not pressed.

As long as the switch is pressed the input will read “0”.



Pull-up resistor

1. https://www.arduino.cc/en/Tutorial/DigitalInputPullup
2. https://www.arduino.cc/en/Tutorial/Foundations/DigitalPins

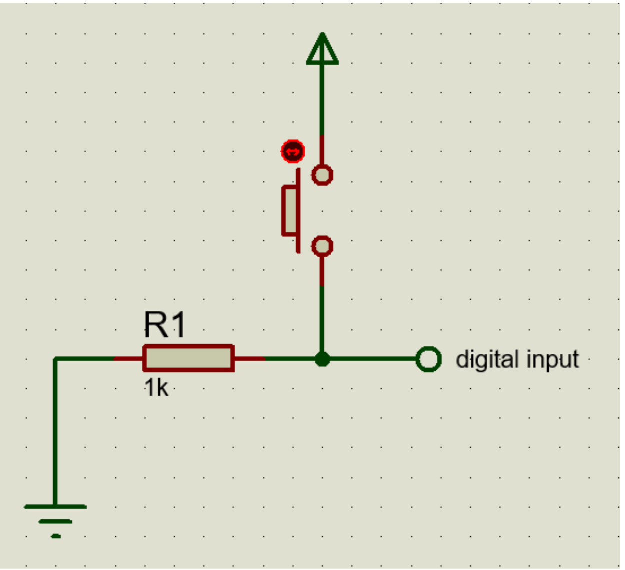
Module_1-2. Pins as inputs

Push-buttons

The push-button can be considered as a normally open switch, which closes for as long as we have pressed it. “0” and “1” can be created from the push-button with the appropriate wiring.

The digital input reads “0” as long as the push-button is not pressed.

As long as the switch is pressed the input will read “1”.



Pull-down resistor

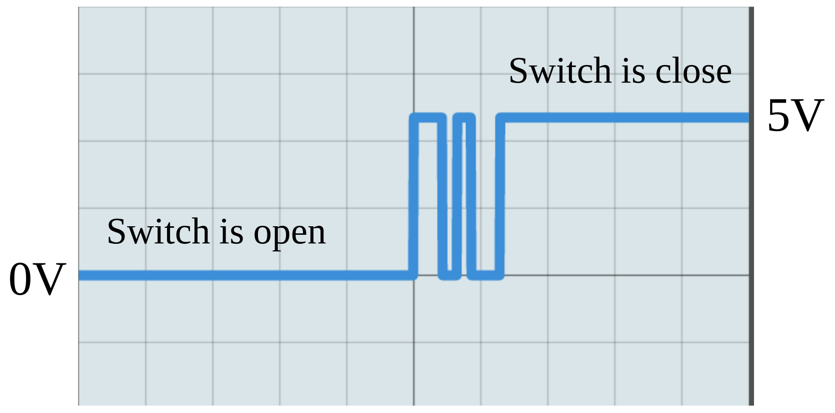
Module_1-2. Pins as inputs

Push-buttons

Mechanical parts, such as switches and push-buttons, cause the digital input signal to bounce.

These bounces result in the input going back and forth between “0” and “1” for a short period of time.

To avoid bounces, an easy and quick solution is to freeze the program for a few milliseconds (approximately 25ms).



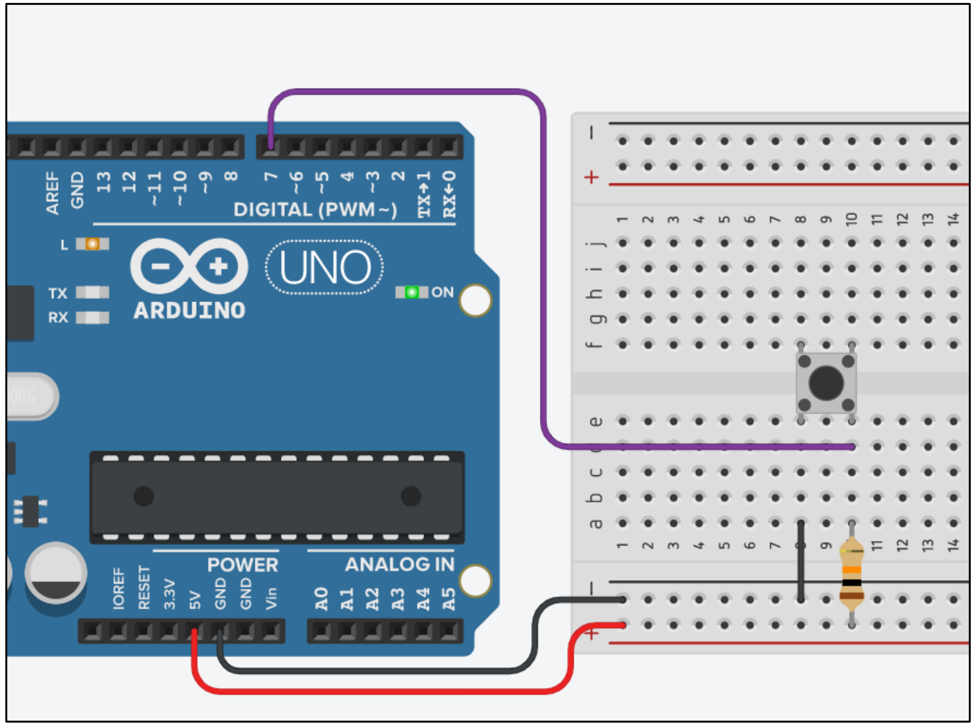
Bounces

1. https://www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/
2. https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce

Module_1-2. Pins as inputs

Example

The example uses the built-in LED (connected to pin 13) and a push-button connected to pin 7. The LED lights up while the push-button is pressed.



Circuit connection

https://www.arduino.cc/en/tutorial/pushbutton

Module_1-2. Pins as inputs

Example

The code:

```
/*  https://www.arduino.cc/en/tutorial/pushbutton
 *  Basic Digital Read
 * ------------------
 *
 * turns on and off a light emitting diode(LED) connected to digital
 * pin 13, when pressing a pushbutton attached to pin 7. It illustrates the
 * concept of Active-Low, which consists in connecting buttons using a
 * 1K to 10K pull-up resistor.
 *
 * Created 1 December 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 */

int ledPin = 13; // choose the pin for the LED
int inPin = 7;   // choose the input pin (for a pushbutton)
int val = 0;     // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT);  // declare LED as output
  pinMode(inPin, INPUT);    // declare pushbutton as input
}

void loop(){
  val = digitalRead(inPin);  // read input value
  if (val == HIGH) {         // check if the input is HIGH (button released)
    digitalWrite(ledPin, LOW);  // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH);  // turn LED ON
  }
}
```

ENGINE Partnership

- Warsaw University of Technology (PL) - *coordinator*
- IHU - International Hellenic University (GR)
- EDUMOTIVA - European Lab for Educational Technology (GR)
- University of Padova (IT)
- University of Applied Sciences in Tarnow (PL)

ENGINE ERASMUS+ 2020-1-PL01-KA226-HE-095653

CONTACT:

www.engined.eu

angelika.tefelska@pw.edu.pl

@projectENGINE1

@EUprojectEngine

Erasmus+

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.