

# ENGINE



Erasmus+

TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING  
IN HIGHER EDUCATION

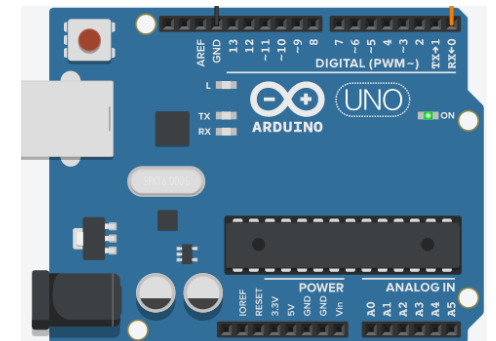
---

## Module\_1-3. Communication & ADC

*Arduino Uno with Tinkercad*

# Contents

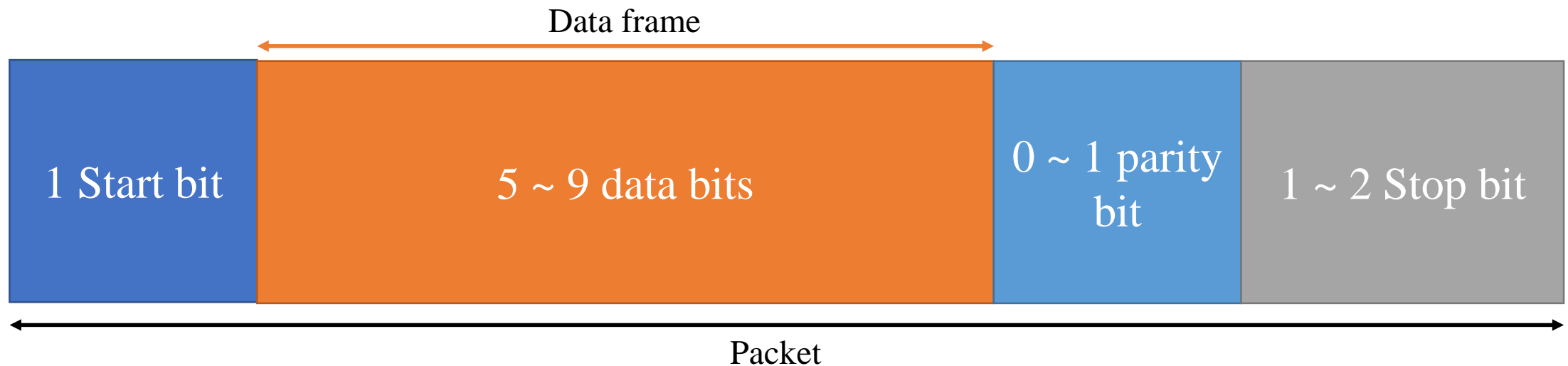
- Serial communication
- Analog to Digital Converter (ADC)
- Programming functions for the Arduino Uno
- Example



# Module\_1-3. Communication and ADC

## Serial Communication

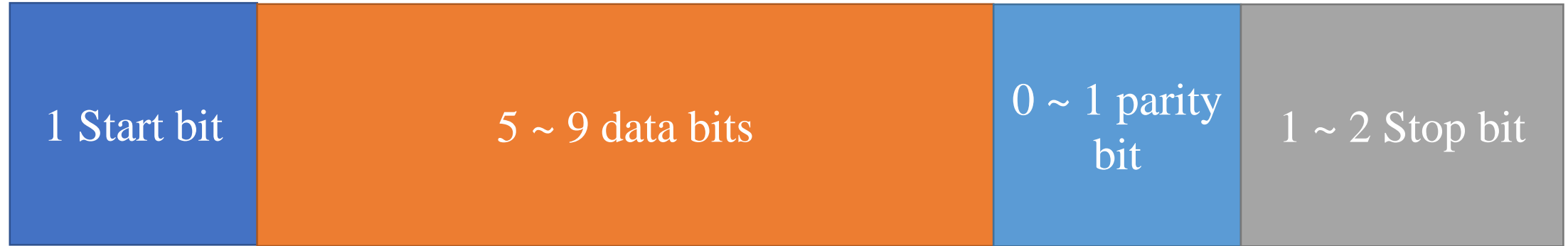
Serial communication is a widespread way of transmitting data between two devices.



1. <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>
2. <https://www.circuitbasics.com/basics-uart-communication/>

# *Module\_1-3. Communication and ADC*

## *Serial Communication*



### Advantages:

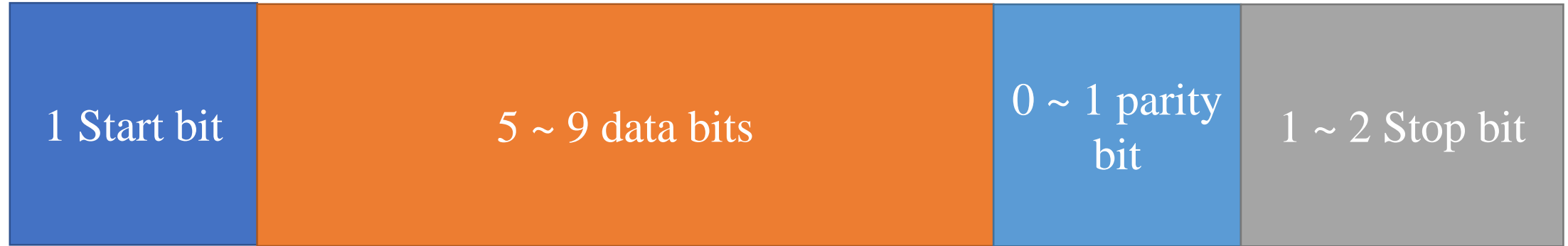
- Communication with just two wires: Rx - Tx
- Error check: parity bit
- Asynchronous communication

1. <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>

2. <https://www.circuitbasics.com/basics-uart-communication/>

# *Module\_1-3. Communication and ADC*

## *Serial Communication*



### Disadvantages:

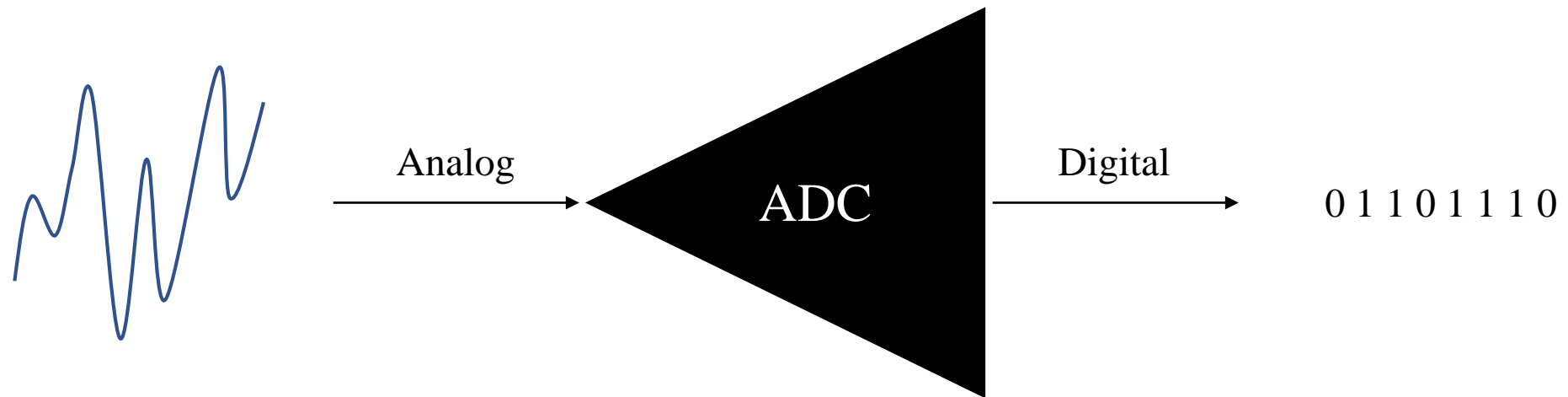
- The maximum data range per packet is just 9 bits
- It does not support multiple devices

1. <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>
2. <https://www.circuitbasics.com/basics-uart-communication/>

# *Module\_1-3. Communication and ADC*

## *Analog to Digital Converter*

The analog-to-digital converter takes a “snapshot” of the unknown analog signal and converts it to bits.



1. <https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html>
2. <http://www.onmyphd.com/?p=analog.digital.converter>
3. <https://dewesoft.com/daq/types-of-adc-converters>

# Module\_1-3. Communication and ADC

## Analog to Digital Converter

A 3-bit ADC has  $2^3 - 1 = 7$  combinations, that is 8 different states (0 ~ 7).

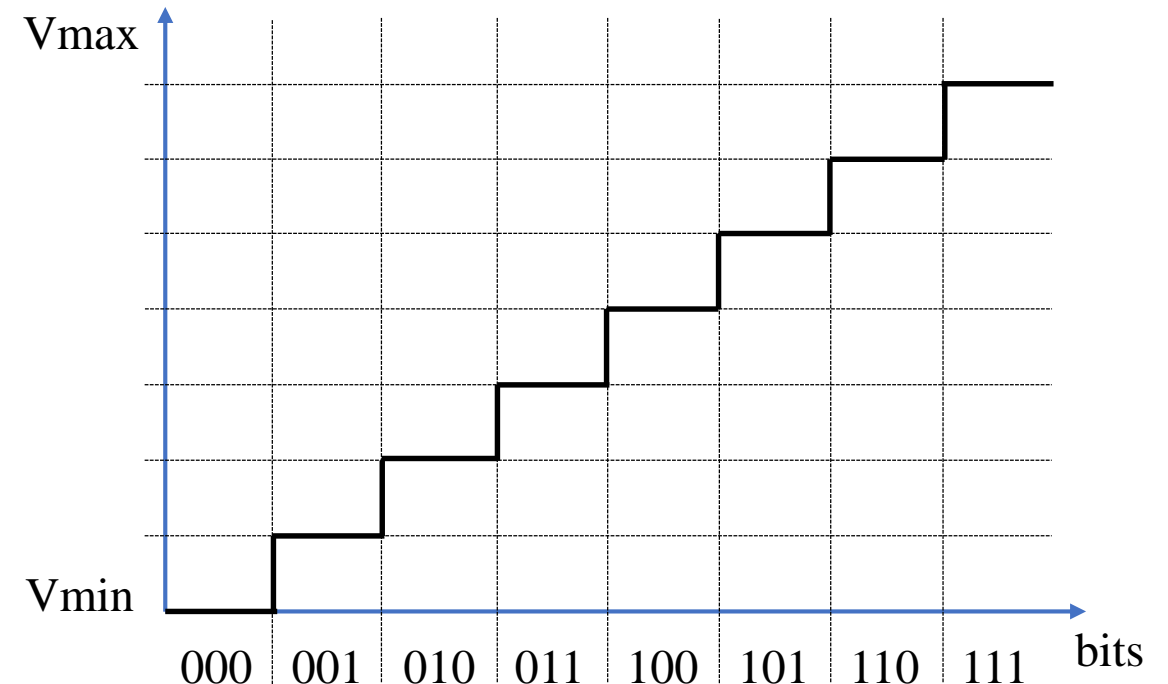
In other words, the ADC can split a voltage range (span) into 8 states. Each state corresponds to 3 bits.

The minimum amount of voltage that an ADC can measure and assign to the 1st bit is

$$LSB = \frac{Span}{2^n - 1}$$

where n are the available converter bits.

*LSB = Least-Significant Bit*



1. <https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html>
2. <http://www.onmyphd.com/?p=analog.digital.converter>
3. <https://dewesoft.com/daq/types-of-adc-converters>

# *Module\_1-3. Communication and ADC Programming functions*

Functions that can be used on the Arduino Uno as we have seen:

- `pinMode(pin, value), delay(value)`
- `digitalRead(pin), digitalWrite(pin, value)`
- `analogWrite(pin, value)`

New functions:

- `Serial.begin(value)`: opens serial port – the value sets the rate for bits per seconds
- `Serial.print()`: prints ASCII characters on the serial port
- `Serial.println()`: same as the previous one. In addition the cursor goes to a new line
- `Serial.available()`: returns the number of characters available for reading on the serial port
- `analogRead(pin)`: reads an analog signal from the pin and returns a 10-bit number

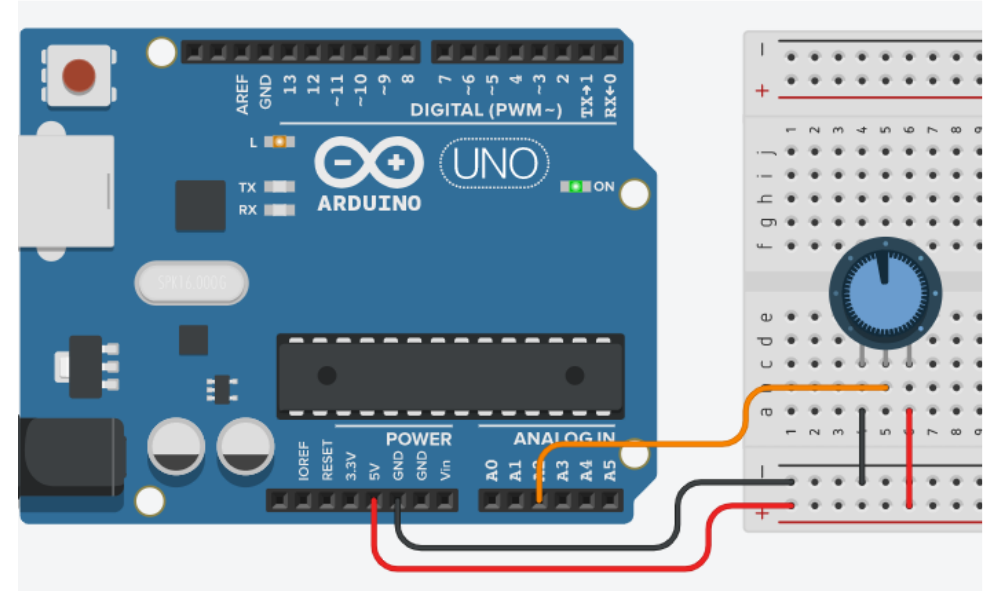
1. <https://www.arduino.cc/reference/en/language/functions/communication/serial/>
2. <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>



# Module\_1-3. Communication and ADC Example

The example uses the built-in LED connected to pin 13 and a potentiometer connected to pin A2.

The LED flashes at a speed set by the potentiometer.



Circuit connection

1. <https://www.arduino.cc/en/tutorial/potentiometer>

# Module\_1-3. Communication and ADC

## Example

The code:

---

```
// https://www.arduino.cc/en/tutorial/potentiometer
/* Analog Read to LED
 * -----
 *
 * turns on and off a light emitting diode(LED) connected to digital
 * pin 13. The amount of time the LED will be on and off depends on
 * the value obtained by analogRead(). In the easiest case we connect
 * a potentiometer to analog pin 2.
 *
 * Created 1 December 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 */
int potPin = 2; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int val = 0; // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}

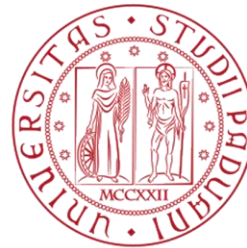
void loop() {
  val = analogRead(potPin); // read the value from the sensor
  digitalWrite(ledPin, HIGH); // turn the ledPin on
  delay(val); // stop the program for some time
  digitalWrite(ledPin, LOW); // turn the ledPin off
  delay(val); // stop the program for some time
}
```

## 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)



INTERNATIONAL  
HELLENIC  
UNIVERSITY



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



## CONTACT:



[www.engined.eu](http://www.engined.eu)



[angelika.tefelska@pw.edu.pl](mailto:angelika.tefelska@pw.edu.pl)



[@projectENGINE1](https://twitter.com/projectENGINE1)



[@EUprojectEngine](https://www.facebook.com/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.