INTRO TO EMBEDDED SYSTEMS WORKSHOP

Tsen Chee Vincent LEUNG YIN KO

AGENDA

- Introduction to Embedded Systems
- Hands on Workshop!
- Hardware familiarization
- Software familiarization
- Actuators and Sensors
- Demo of applications

WHAT ARE EMBEDDED SYSTEMS

WHAT IS IT

- “Dumb” Devices
- Smart Devices
- Internet of Things
- Robotics

WHERE IS IT

- Anything “Smart”
- Microwaves/Fridges
- Drones/Self Driving Cars
- NASA Rovers
- Biochips

PROVIDED SUPPLIES

1. ARDUINO UNO/MEGA (1x)
2. USB CABLE (1x)
3. BREADBOARD (1x)
4. HOOKUP WIRE (1m Black, 1m Red)
5. LED (5x)
6. BUTTONS (2x)
7. RESISTORS (7x 330Ω, 2x 1kΩ)
8. CAPACITORS (2x 33nF, 2x 100nF, 2x 470µF)

Hardware

Overview

☐ Arduino Datasheet

☐ Connection

☐ Arduino IDE Installation

☐ “Hello World”

☐ Flashing

☐ Serial Monitor

ACTIVITY 1
SETUP THE BRAIN

Task: Write “Hello World” on the Serial Monitor
Outcome

- Microcontroller documentation
- Arduino IDE Setup
- Hardware + Software familiarization

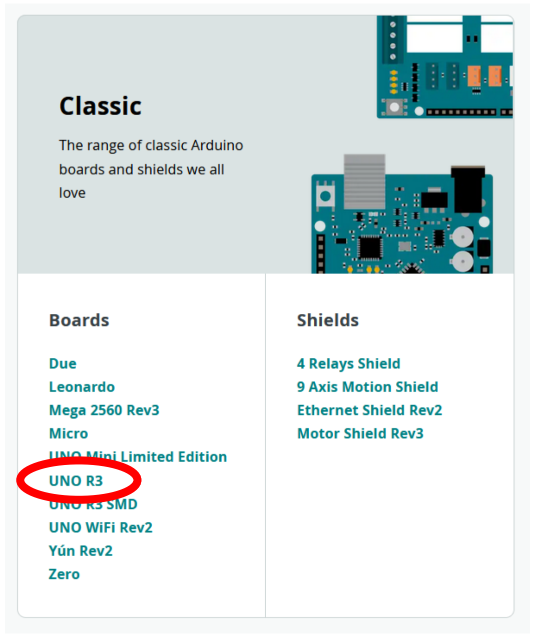
ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

- ☐ Arduino Datasheet
- ☐ Connection
- ☐ Arduino IDE Installation
- ☐ "Hello World"
- ☐ Flashing
- ☐ Serial Monitor

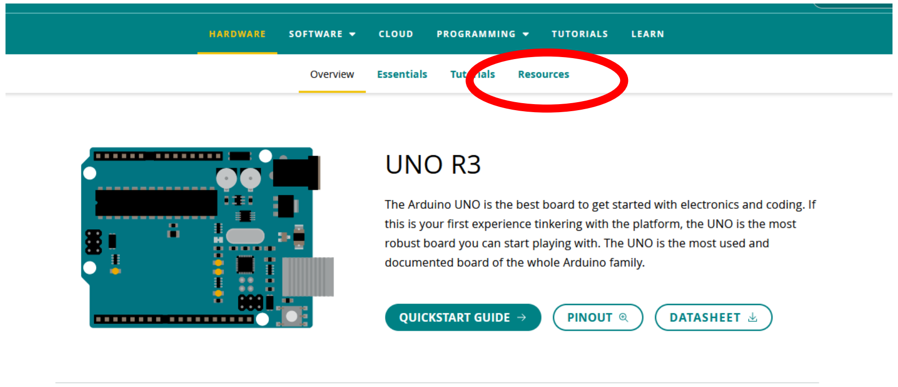
1. Download the Arduino Datasheet (https://docs.arduino.cc/hardware/uno-rev3)



Classic

The range of classic Arduino boards and shields we all love

Boards	Shields
Due	4 Relays Shield
Leonardo	9 Axis Motion Shield
Mega 2560 Rev3	Ethernet Shield Rev2
Micro	Motor Shield Rev3
UNO Mini Limited Edition	
UNO R3	
UNO R3 SMD	
UNO WiFi Rev2	
Yún Rev2	
Zero	



UNO R3

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

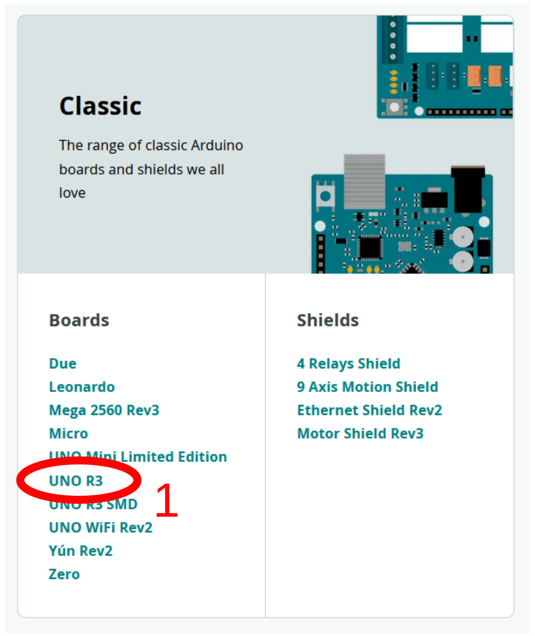
ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

- ☐ Arduino Datasheet
- ☐ Connection
- ☐ Arduino IDE Installation
- ☐ "Hello World"
- ☐ Flashing
- ☐ Serial Monitor

1. Download the Arduino Datasheet (https://docs.arduino.cc/hardware/uno-rev3)



Classic

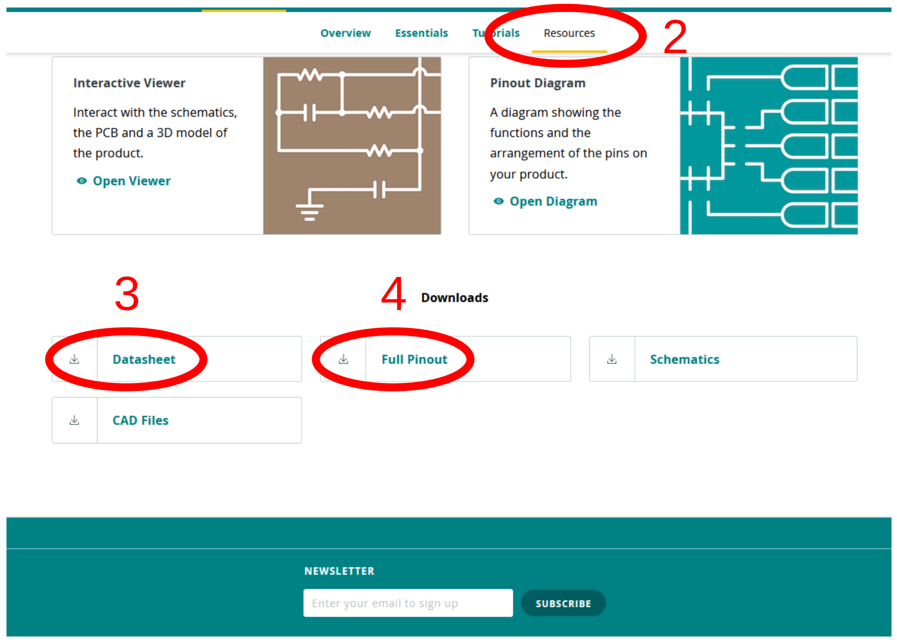
The range of classic Arduino boards and shields we all love

Boards

Due
Leonardo
Mega 2560 Rev3
Micro
UNO Mini Limited Edition
UNO R3
UNO R3 SMD
UNO WiFi Rev2
Yún Rev2
Zero

Shields

4 Relays Shield
9 Axis Motion Shield
Ethernet Shield Rev2
Motor Shield Rev3



Interactive Viewer

Interact with the schematics, the PCB and a 3D model of the product.

Pinout Diagram

A diagram showing the functions and the arrangement of the pins on your product.

Downloads

ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

☑ Arduino Datasheet

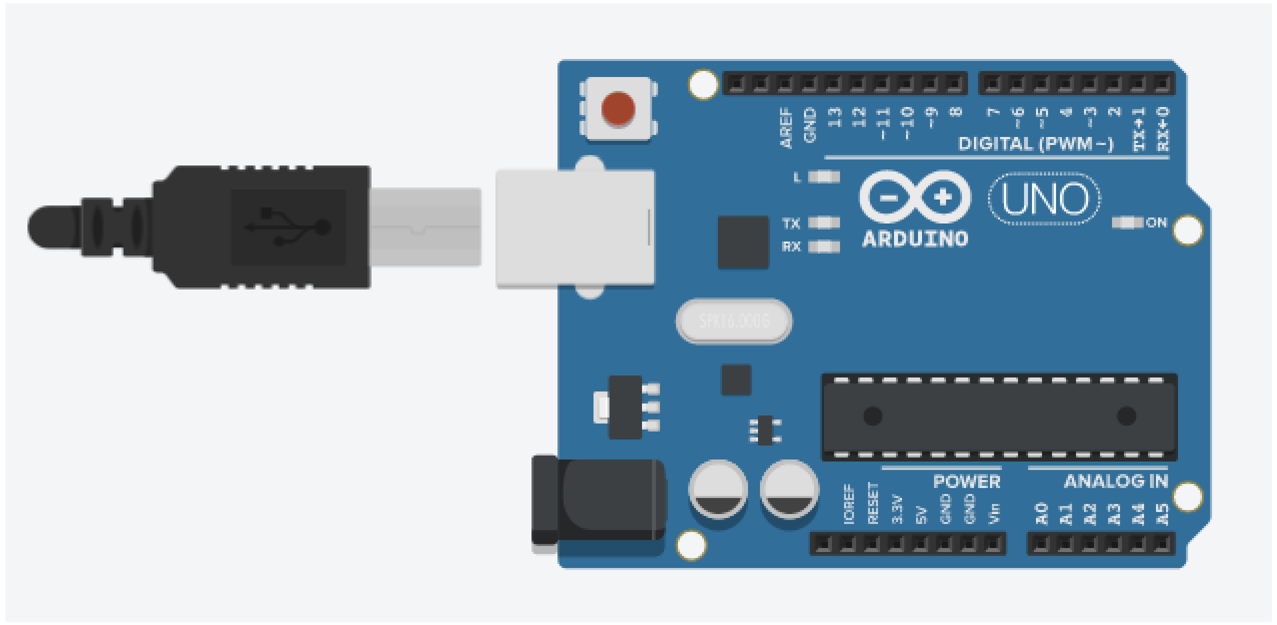
☐ Connection

☐ Arduino IDE Installation

☐ “Hello World”

☐ Flashing

☐ Serial Monitor



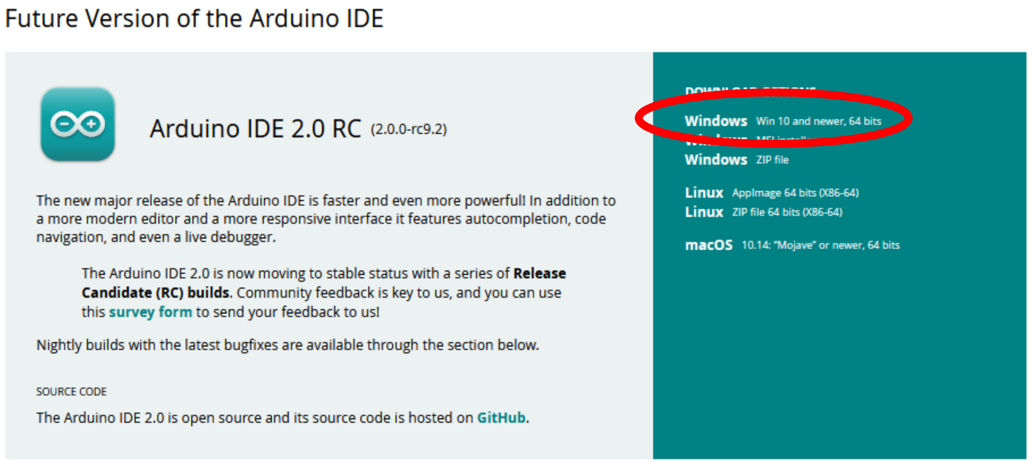
ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

- ☑ Arduino Datasheet
- ☑ Connection
- ☐ Arduino IDE Installation
- ☐ "Hello World"
- ☐ Flashing
- ☐ Serial Monitor

1. Download the Arduino IDE v2.x (https://docs.arduino.cc/software/ide-v2)
2. Install & Launch


Future Version of the Arduino IDE

Arduino IDE 2.0 RC (2.0.0-rc9.2)

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

The Arduino IDE 2.0 is now moving to stable status with a series of **Release Candidate (RC) builds**. Community feedback is key to us, and you can use this survey form to send your feedback to us!

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

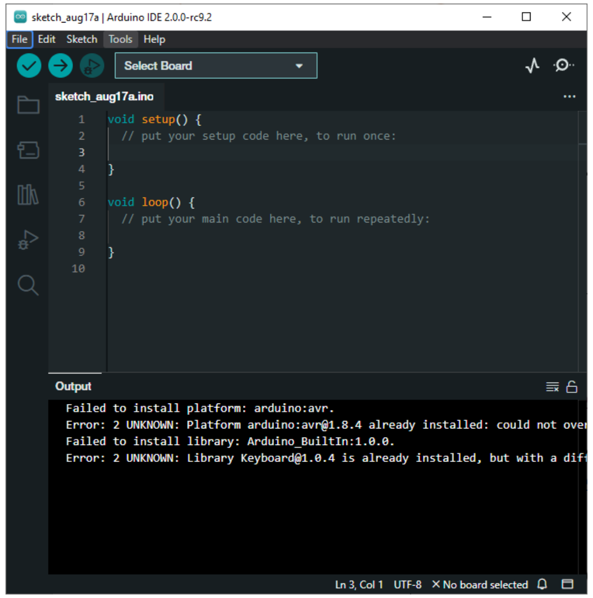
The Arduino IDE 2.0 is open source and its source code is hosted on GitHub.

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits


sketch_aug17a | Arduino IDE 2.0.0-rc9.2

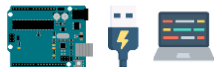
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
Failed to install platform: arduino:avr.
Error: 2 UNKNOWN: Platform arduino:avr@1.8.4 already installed: could not over
Failed to install library: Arduino_BuiltIn:1.0.0.
Error: 2 UNKNOWN: Library Keyboard@1.0.4 is already installed, but with a dif
```

Ln 3, Col 1 UTF-8 ✕ No board selected

ACTIVITY 1
SETUP THE BRAIN

Hardware



Overview

☑ Arduino Datasheet

☑ Connection

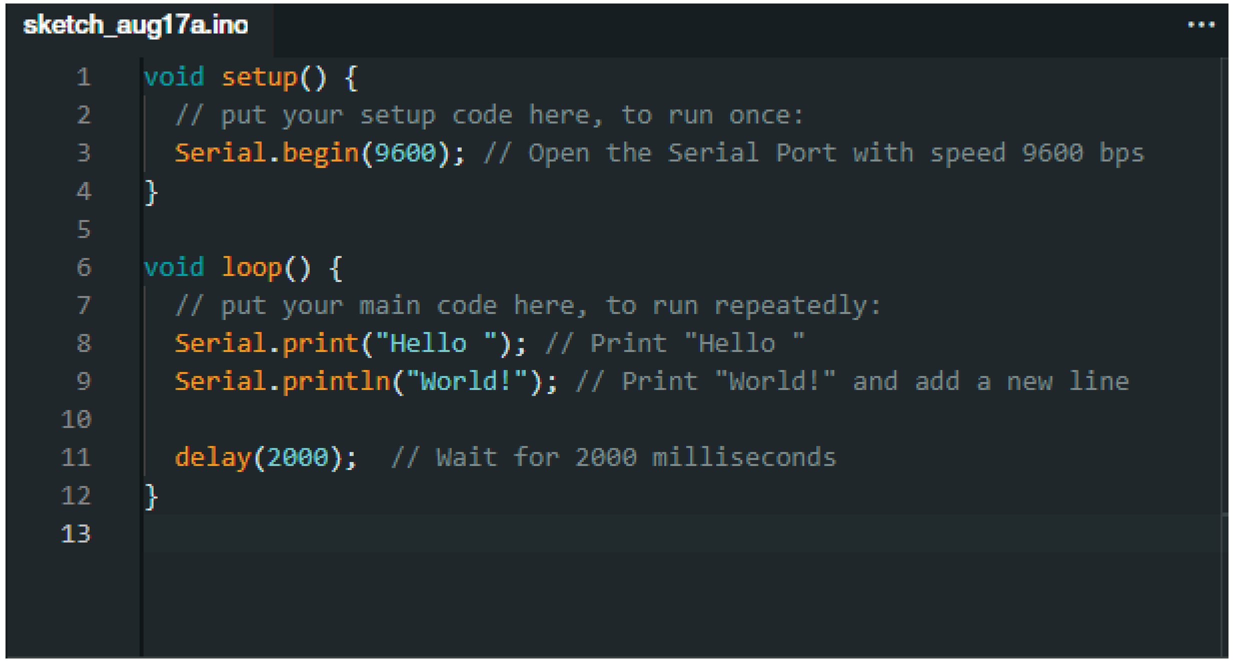
☑ Arduino IDE Installation

☐ “Hello World”

☐ Flashing

☐ Serial Monitor

1. Code a simple “Hello World”



sketch_aug17a.ino

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with speed 9600 bps
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hello "); // Print "Hello "
  Serial.println("World!"); // Print "World!" and add a new line

  delay(2000);  // Wait for 2000 milliseconds
}

```

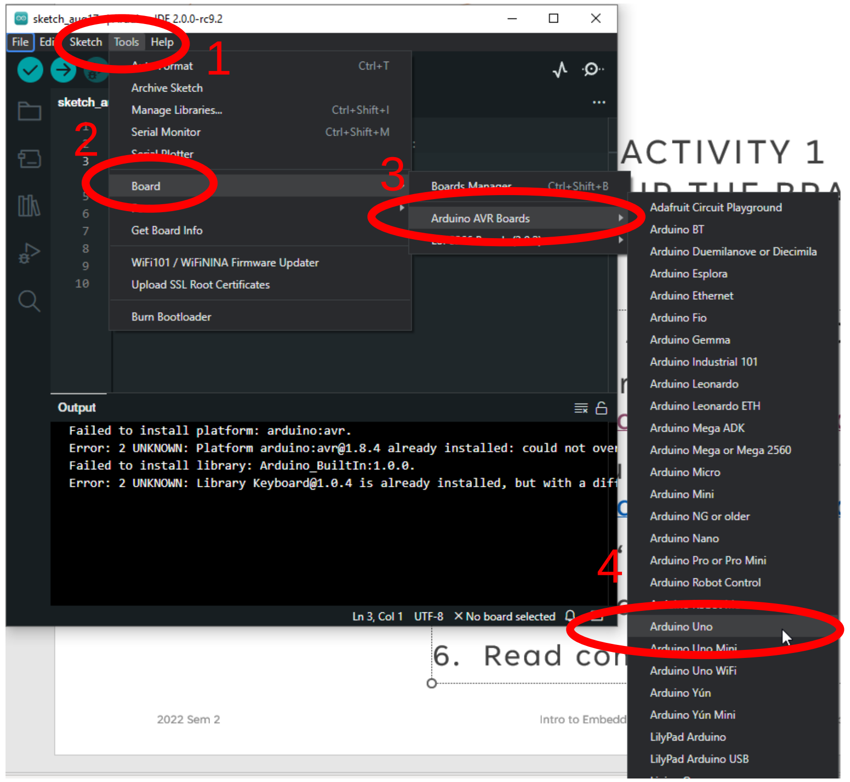
ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

- ☑ Arduino Datasheet
- ☑ Connection
- ☑ Arduino IDE Installation
- ☑ “Hello World”
- ☐ Flashing
- ☐ Serial Monitor

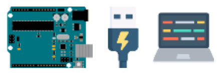
1. Tools -> Board -> Arduino AVR Boards -> Arduino Uno



```
Failed to install platform: arduino:avr.
Error: 2 UNKNOWN: Platform arduino:avr@1.8.4 already installed: could not over
Failed to install library: Arduino_BuiltIn:1.0.0.
Error: 2 UNKNOWN: Library Keyboard@1.0.4 is already installed, but with a dif
```

ACTIVITY 1
SETUP THE BRAIN

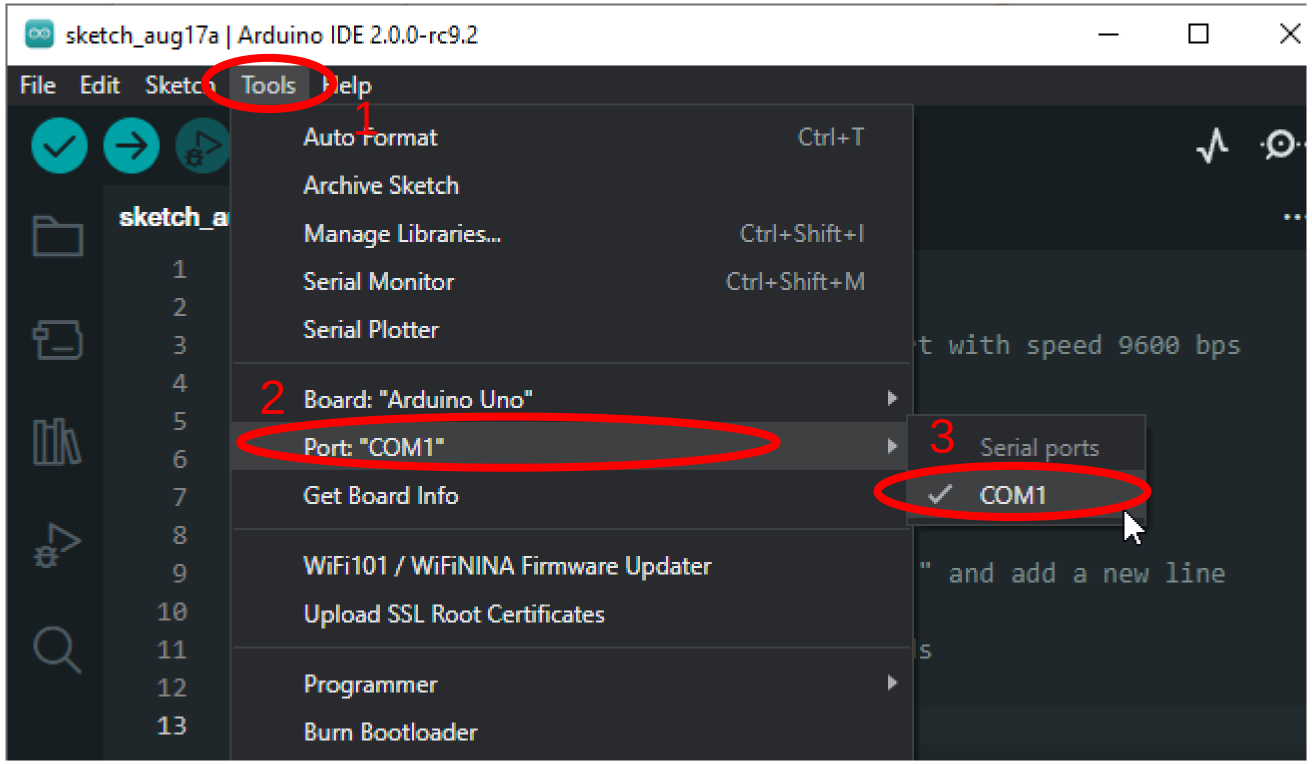
Hardware



Overview

- ☑ Arduino Datasheet
- ☑ Connection
- ☑ Arduino IDE Installation
- ☑ "Hello World"
- ☐ Flashing
- ☐ Serial Monitor

1. Tools -> Port -> COM[X]



sketch_aug17a | Arduino IDE 2.0.0-rc9.2

File Edit Sketch Tools Help

1 Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
2 Board: "Arduino Uno"
Port: "COM1"
Get Board Info
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Programmer
Burn Bootloader

3 Serial ports
✓ COM1

```
t with speed 9600 bps
" and add a new line
s
```

ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

☑ Arduino Datasheet

☑ Connection

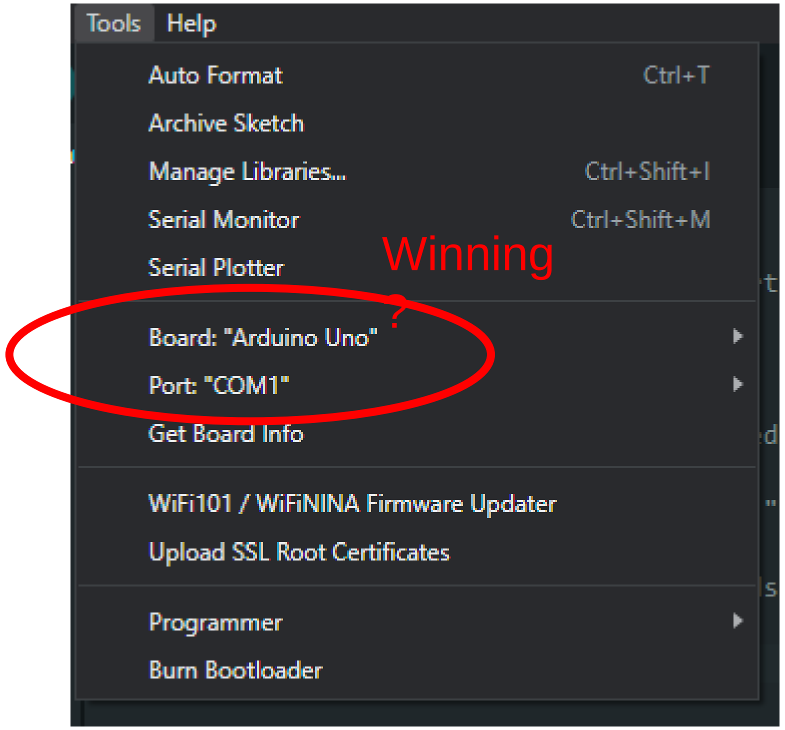
☑ Arduino IDE Installation

☑ "Hello World"

☐ Flashing

☐ Serial Monitor

1. Board "Arduino Uno", Port "COM[X]"



Tools Help

Auto Format Ctrl+T

Archive Sketch

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter

Board: "Arduino Uno"

Port: "COM1"

Get Board Info

WiFi101 / WiFiNINA Firmware Updater

Upload SSL Root Certificates

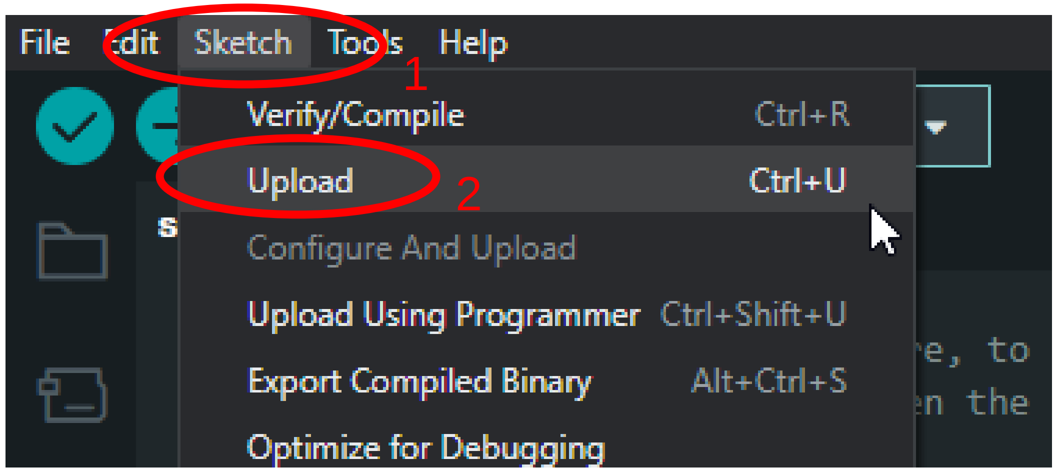
Programmer

Burn Bootloader

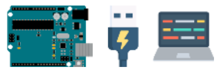
Winning ?

ACTIVITY 1
SETUP THE BRAIN

1. Flash -> Upload Files (Aka“Flashing”)



Hardware

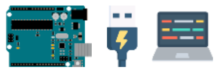


Overview

☑ Arduino Datasheet
☑ Connection
☑ Arduino IDE Installation
☑ “Hello World”
☐ Flashing
☐ Serial Monitor

ACTIVITY 1
SETUP THE BRAIN

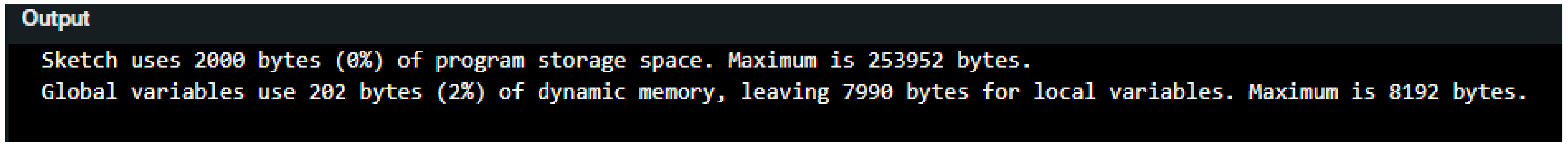
Hardware



Overview

- ☑ Arduino Datasheet
- ☑ Connection
- ☑ Arduino IDE Installation
- ☑ "Hello World"
- ☐ Flashing
- ☐ Serial Monitor

1. Notice output console magicks + details



```
Sketch uses 2000 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 202 bytes (2%) of dynamic memory, leaving 7990 bytes for local variables. Maximum is 8192 bytes.
```

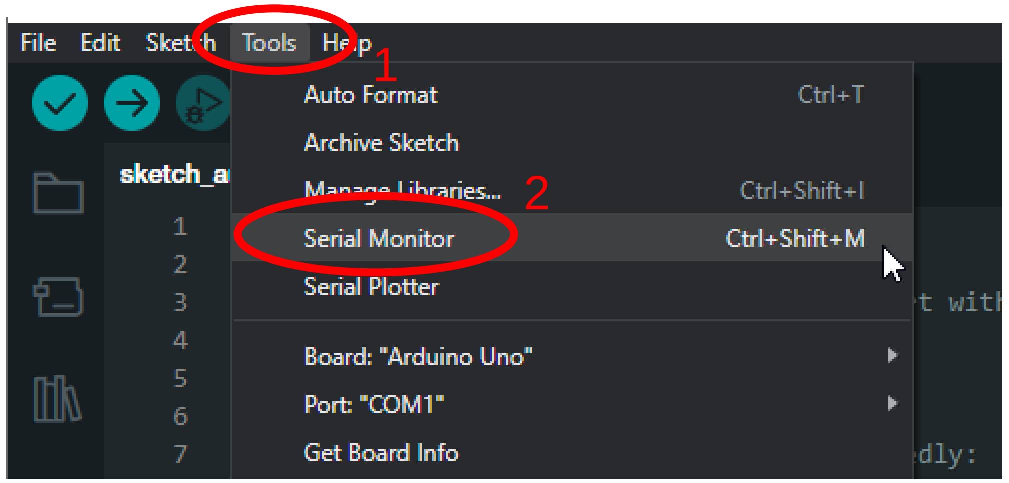
ACTIVITY 1
SETUP THE BRAIN

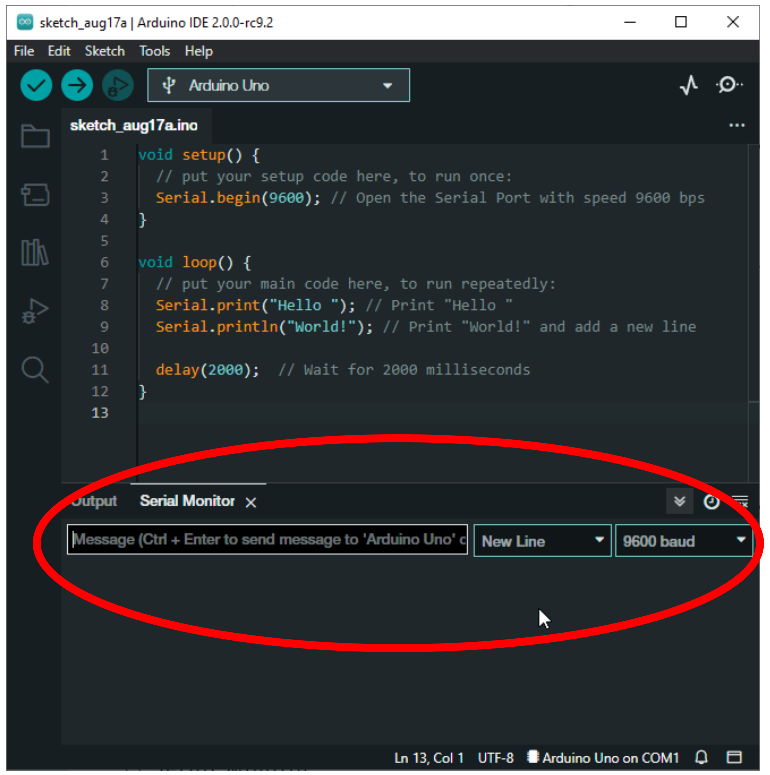
Hardware

Overview

- ☑ Arduino Datasheet
- ☑ Connection
- ☑ Arduino IDE Installation
- ☑ “Hello World”
- ☑ Flashing
- ☐ Serial Monitor

1. Tools -> Serial Monitor
2. What do you see?




sketch_aug17a | Arduino IDE 2.0.0-rc9.2
File Edit Sketch Tools Help
Arduino Uno
sketch_aug17a.ino
Output
Serial Monitor
Message (Ctrl + Enter to send message to 'Arduino Uno' c

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with speed 9600 bps
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hello "); // Print "Hello "
  Serial.println("World!"); // Print "World!" and add a new line

  delay(2000);  // Wait for 2000 milliseconds
}
```

New Line | 9600 baud

Ln 13, Col 1   UTF-8   Arduino Uno on COM1

ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

☑ Arduino Datasheet

☑ Connection

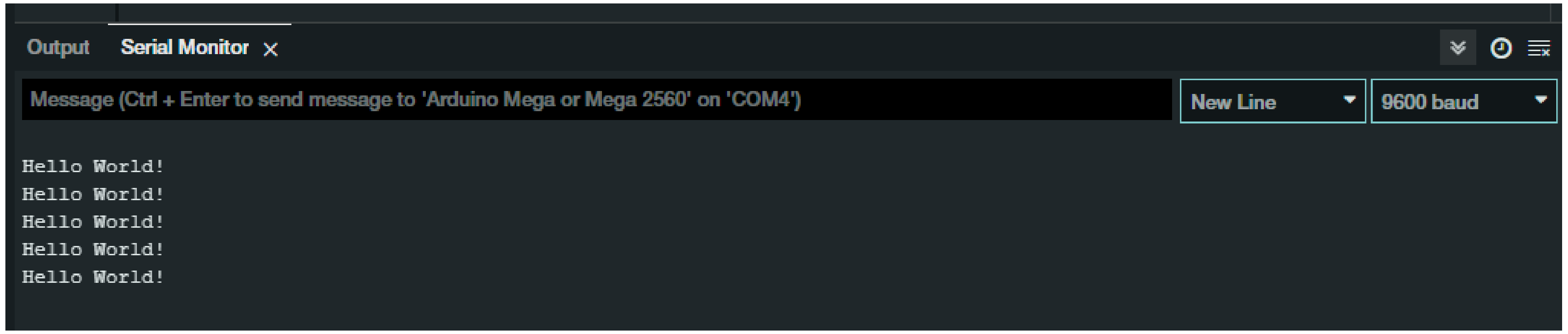
☑ Arduino IDE Installation

☑ "Hello World"

☑ Flashing

☐ Serial Monitor

1. Tools -> Serial Monitor
2. What do you see? Hello World!



Message (Ctrl + Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')

New Line

9600 baud

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

ACTIVITY 1
SETUP THE BRAIN

Hardware

Overview

☑ Arduino Datasheet

☑ Connection

☑ Arduino IDE Installation

☑ “Hello World”

☑ Flashing

☑ Serial Monitor

Task: Write “Hello World” on the Serial Monitor

Outcome

- Microcontroller documentation
- Arduino IDE Setup
- Hardware + Software familiarization

Recap

- Arduino is a Microcontroller/the Brains
- Programmable - “Flashable”
- Can communicate

Hardware

Overview

☐ Input and Output

☐ Analog/Digital GPIO

☐ Turn ON/OFF Built-in LED

☐ Blink SOS

☐ Physical Limitations (Datasheets!)

ACTIVITY 2
BLINK FOR HELP

Task: Blink the built-in LED to morse code (SOS)
Outcome

- Microcontroller documentation
- Hardware + Software familiarization
- Output / Actuator intro

BREAK TIME
RESUME AT:

Hardware

Overview

☐ Input and Output

☐ Analog/Digital GPIO

☐ Turn ON/OFF Built-in LED

☐ Blink SOS

☐ Physical Limitations (Datasheets!)

ACTIVITY 2
BLINK FOR HELP

Task: Blink the built-in LED to morse code (SOS)
Outcome

- Microcontroller documentation
- Hardware + Software familiarization
- Output / Actuator intro

ACTIVITY 2
BLINK FOR HELP

1. Communication? Input? Output?
2. Receive? Send?

Hardware

Overview

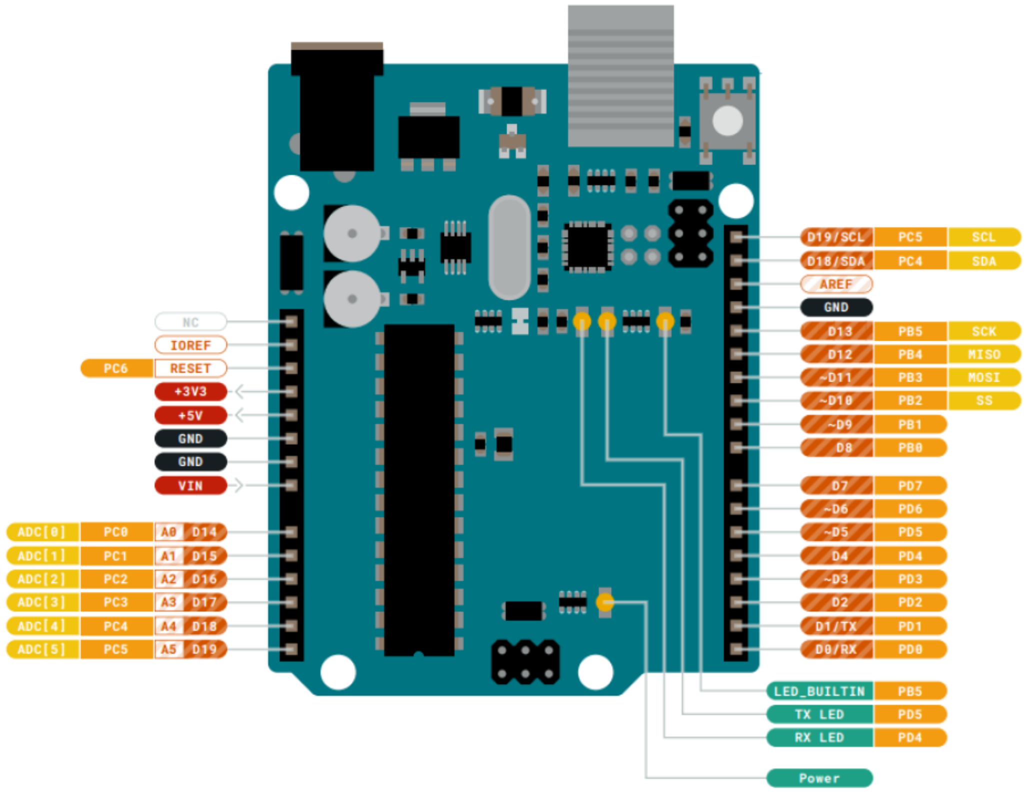
☐ Input and Output

☐ Analog/Digital GPIO

☐ Turn ON/OFF Built-in LED

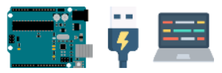
☐ Blink SOS

☐ Physical Limitations (Datasheets!)



ACTIVITY 2
BLINK FOR HELP

Hardware



Overview

☑ Input and Output

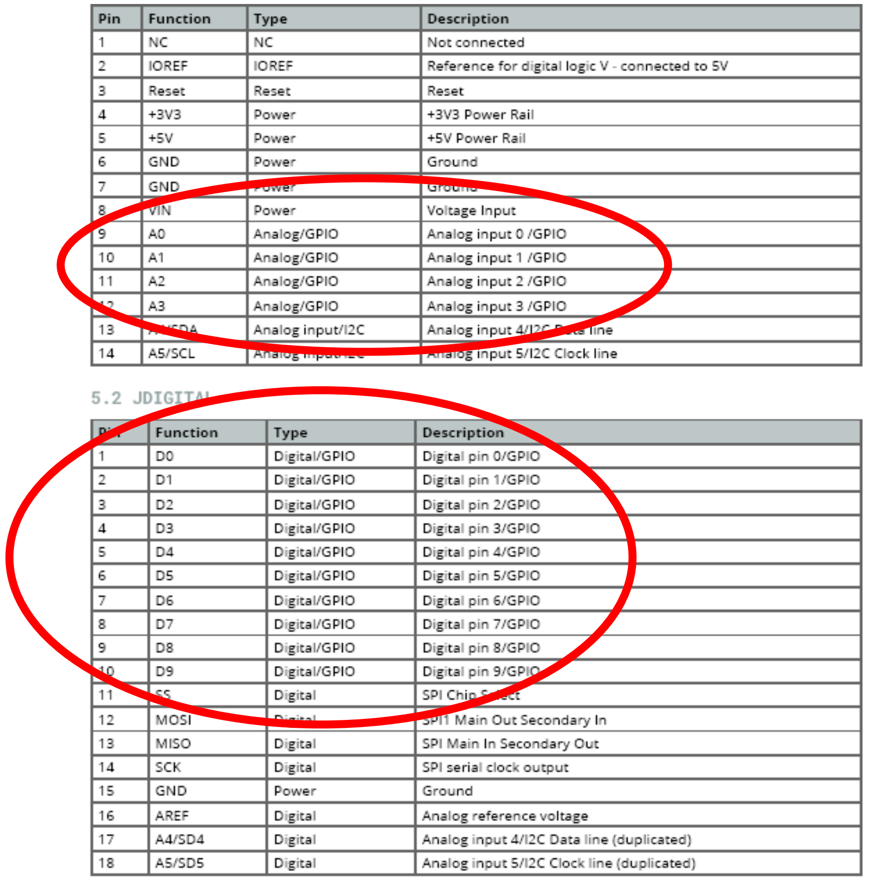
☐ Analog/Digital GPIO

☐ Turn ON/OFF Built-in LED

☐ Blink SOS

☐ Physical Limitations (Datasheets!)

1. General Purpose Input Output (GPIO)
2. Both Analog GPIO and Digital GPIO



Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

ACTIVITY 2
BLINK FOR HELP

Hardware

Overview

☑ Input and Output

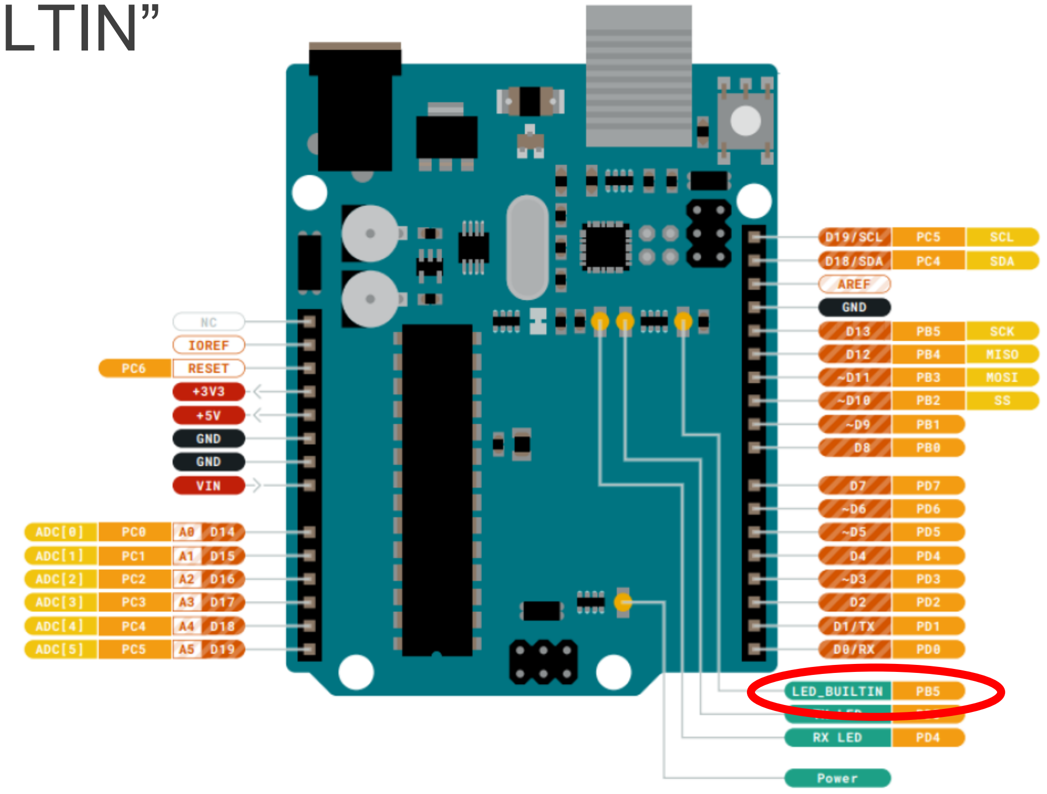
☑ Analog/Digital GPIO

☐ Turn ON/OFF Built-in LED

☐ Blink SOS

☐ Physical Limitations (Datasheets!)

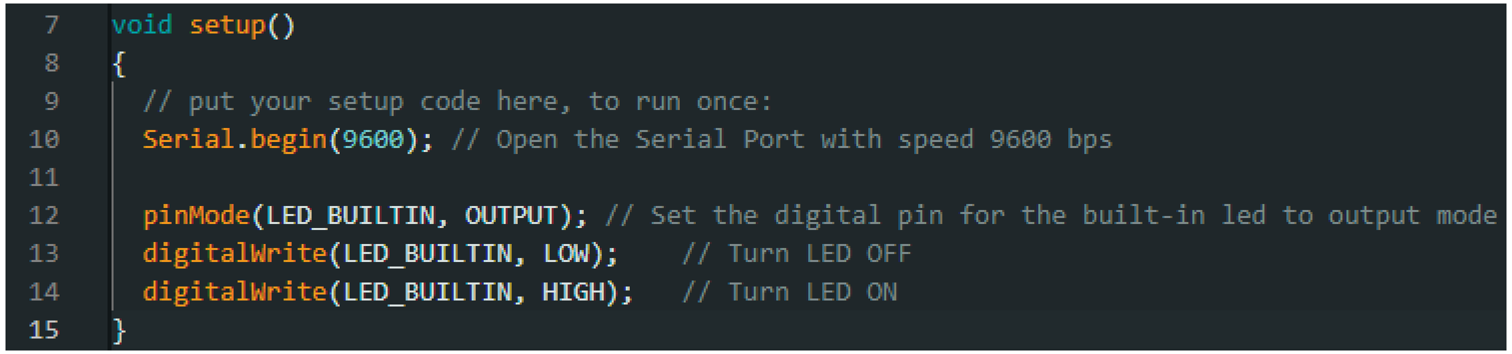
1. Turn ON/OFF LED, Use Digital GPIO!
2. We'll use "LED_BUILTIN"



ACTIVITY 2
BLINK FOR HELP

Hardware

1. Arduino IDE, File -> New
2. Set “LED_BUILTIN” to OUTPUT mode!
3. Turn it OFF and ON



```
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with speed 9600 bps

  pinMode(LED_BUILTIN, OUTPUT); // Set the digital pin for the built-in led to output mode
  digitalWrite(LED_BUILTIN, LOW);    // Turn LED OFF
  digitalWrite(LED_BUILTIN, HIGH);   // Turn LED ON
}
```

1. Flash!

Overview

- ☑ Input and Output
- ☑ Analog/Digital GPIO
- ☐ Turn ON/OFF Built-in LED
- ☐ Blink SOS
- ☐ Physical Limitations (Datasheets!)

ACTIVITY 2
BLINK FOR HELP

Hardware

Overview

☑ Input and Output

☑ Analog/Digital GPIO

☑ Turn ON/OFF Built-in LED

☐ Blink SOS

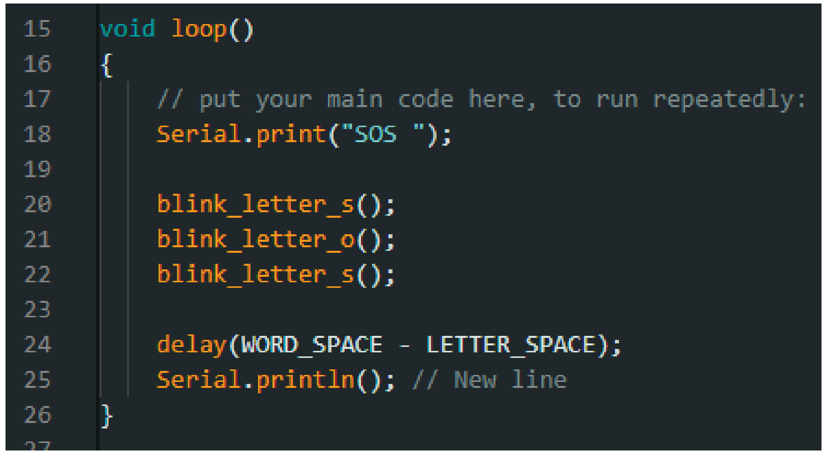
☐ Physical Limitations (Datasheets!)

1. Blink SOS morse code (· · ·----· · ·)
2. Use functions

Morse Code

Dot	100ms
Dash	300ms
Between Dots and Dashes	100ms
Between Letters	300ms
Between Words	700ms

S	**. . .**
O	**---**
S	**. . .**



```
void loop()
{
    // put your main code here, to run repeatedly:
    Serial.print("SOS ");

    blink_letter_s();
    blink_letter_o();
    blink_letter_s();

    delay(WORD_SPACE - LETTER_SPACE);
    Serial.println(); // New line
}
```

ACTIVITY 2
BLINK FOR HELP

Morse Code

Dot	100ms
Dash	300ms
Between Dots and Dashes	100ms
Between Letters	300ms
Between Words	700ms

S	...
O	---
S	...

Hardware

Overview

☑ Input and Output

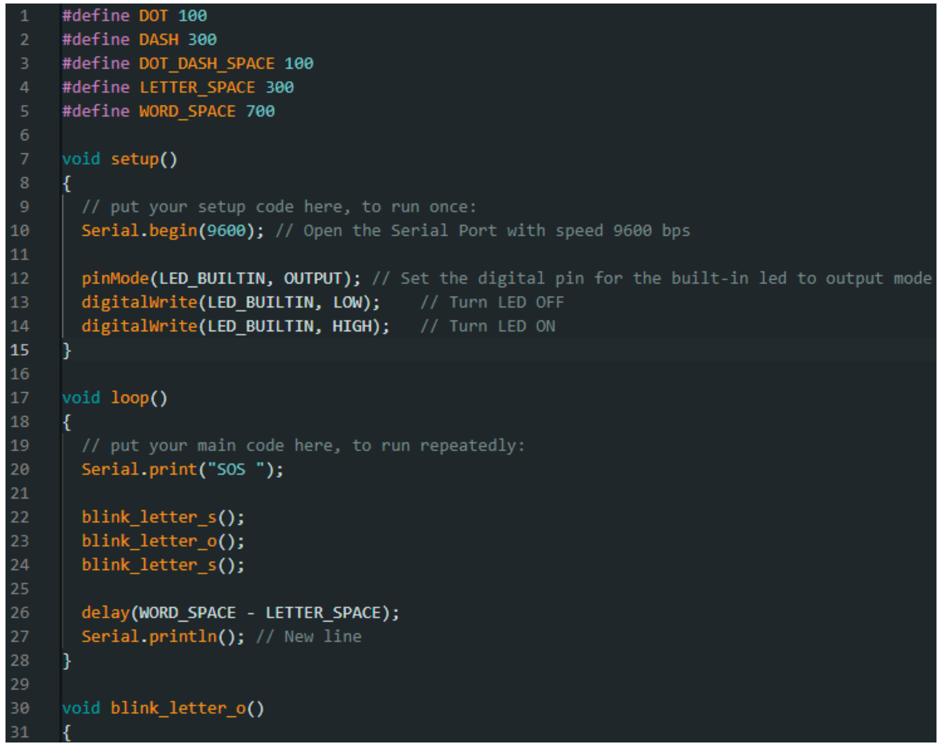
☑ Analog/Digital GPIO

☑ Turn ON/OFF Built-in LED

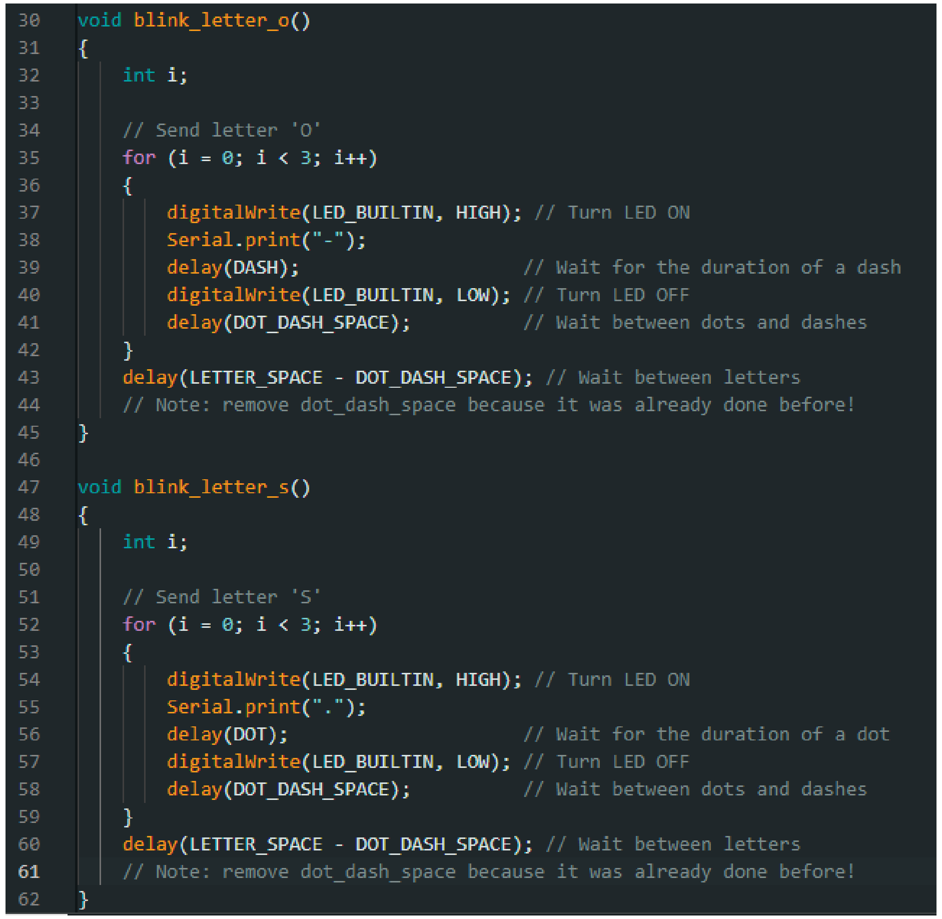
☐ Blink SOS

☐ Physical Limitations (Datasheets!)

1. Flash!



```
#define DOT 100
#define DASH 300
#define DOT_DASH_SPACE 100
#define LETTER_SPACE 300
#define WORD_SPACE 700

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with speed 9600 bps

  pinMode(LED_BUILTIN, OUTPUT); // Set the digital pin for the built-in led to output mode
  digitalWrite(LED_BUILTIN, LOW);    // Turn LED OFF
  digitalWrite(LED_BUILTIN, HIGH);   // Turn LED ON
}

void loop()
{
  // put your main code here, to run repeatedly:
  Serial.print("SOS ");

  blink_letter_s();
  blink_letter_o();
  blink_letter_s();

  delay(WORD_SPACE - LETTER_SPACE);
  Serial.println(); // New line
}

void blink_letter_o()
{
```



```
void blink_letter_o()
{
    int i;

    // Send letter 'O'
    for (i = 0; i < 3; i++)
    {
        digitalWrite(LED_BUILTIN, HIGH); // Turn LED ON
        Serial.print("-");
        delay(DASH);                     // Wait for the duration of a dash
        digitalWrite(LED_BUILTIN, LOW); // Turn LED OFF
        delay(DOT_DASH_SPACE);           // Wait between dots and dashes
    }
    delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
    // Note: remove dot_dash_space because it was already done before!
}

void blink_letter_s()
{
    int i;

    // Send letter 'S'
    for (i = 0; i < 3; i++)
    {
        digitalWrite(LED_BUILTIN, HIGH); // Turn LED ON
        Serial.print(".");
        delay(DOT);                      // Wait for the duration of a dot
        digitalWrite(LED_BUILTIN, LOW); // Turn LED OFF
        delay(DOT_DASH_SPACE);           // Wait between dots and dashes
    }
    delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
    // Note: remove dot_dash_space because it was already done before!
}
```

ACTIVITY 2
BLINK FOR HELP

Hardware

Overview

☑ Input and Output

☑ Analog/Digital GPIO

☑ Turn ON/OFF Built-in LED

☑ Blink SOS

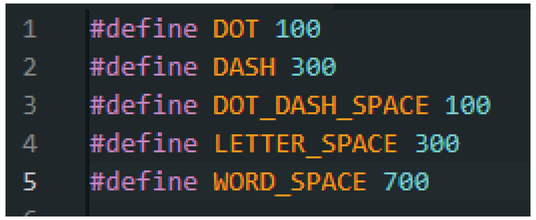
☐ Physical Limitations (Datasheets!)

1. Try reducing the time by a factor of 10x
2. Reduce it by a factor of 100x
3. Does it still blink?
4. Is it still bright?

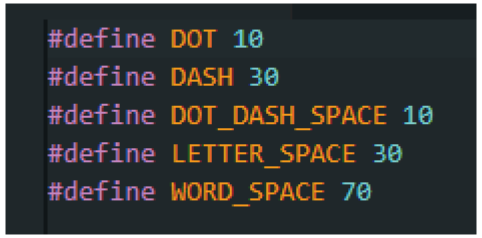
Morse Code

Dot	100ms
Dash	300ms
Between Dots and Dashes	100ms
Between Letters	300ms
Between Words	700ms

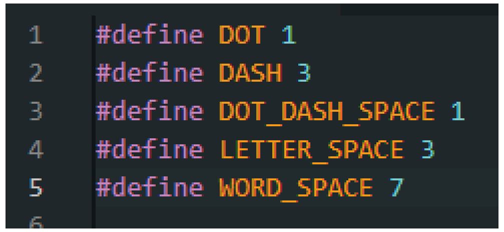
```
S    ...
O    ---
S    ...
```



```
#define DOT 100
#define DASH 300
#define DOT_DASH_SPACE 100
#define LETTER_SPACE 300
#define WORD_SPACE 700
```



```
#define DOT 10
#define DASH 30
#define DOT_DASH_SPACE 10
#define LETTER_SPACE 30
#define WORD_SPACE 70
```



```
#define DOT 1
#define DASH 3
#define DOT_DASH_SPACE 1
#define LETTER_SPACE 3
#define WORD_SPACE 7
```

ACTIVITY 2
BLINK FOR HELP

Hardware

Overview

☑ Input and Output

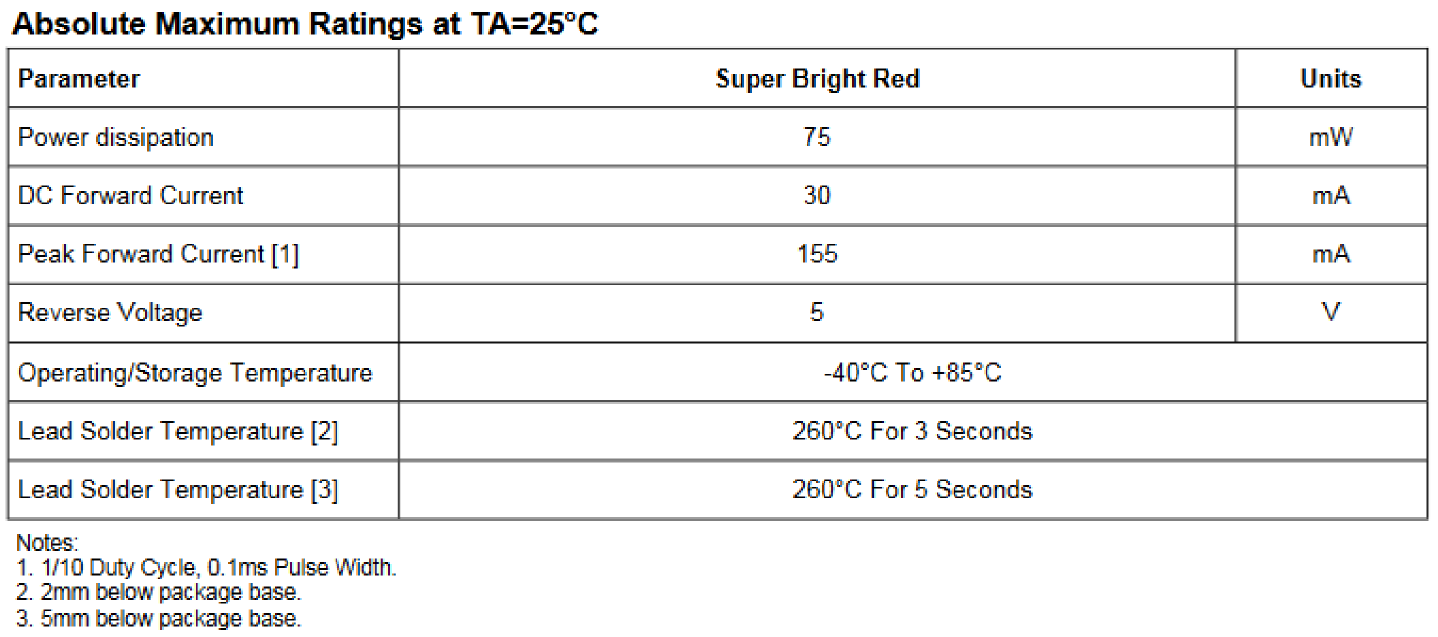
☑ Analog/Digital GPIO

☑ Turn ON/OFF Built-in LED

☑ Blink SOS

☐ Physical Limitations (Datasheets!)

1. Typical LED Datasheet



Absolute Maximum Ratings at TA=25°C

Parameter	Super Bright Red	Units
Power dissipation	75	mW
DC Forward Current	30	mA
Peak Forward Current [1]	155	mA
Reverse Voltage	5	V
Operating/Storage Temperature	-40°C To +85°C	
Lead Solder Temperature [2]	260°C For 3 Seconds	
Lead Solder Temperature [3]	260°C For 5 Seconds	

Notes:
1. 1/10 Duty Cycle, 0.1ms Pulse Width.
2. 2mm below package base.
3. 5mm below package base.

Further reading: Pulse Width Modulation (PWM)

Hardware

Overview

☑ Input and Output

☑ Analog/Digital GPIO

☑ Turn ON/OFF Built-in LED

☑ Blink SOS

☑ Physical Limitations (Datasheets!)

ACTIVITY 2
BLINK FOR HELP

Task: Blink the built-in LED to morse code (SOS)
Outcome

- Microcontroller documentation
- Hardware + Software familiarization
- Output / Actuator intro

Recap

- A/O GPIO
- Knowledge in documentation and/or datasheets!
- Know the limitations (software and hardware)

Hardware

Overview

☐ Connect External Hardware

☐ Modify SOS

☐ Send command via Serial

ACTIVITY 3 LIGHTBRINGER

Task: Blink an external LED to morse code (SOS) on command
Outcome

- Schematic reading (basic)
- Hardware + Software familiarization
- Serial & Serial Monitor

BREAK TIME
RESUME AT:

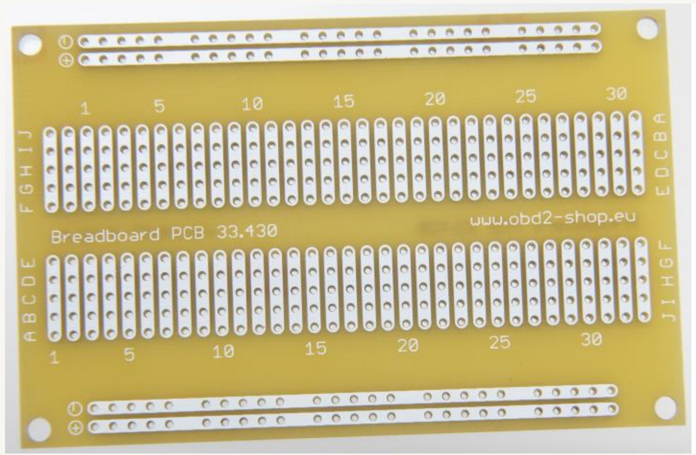
WORKING WITH ELECTRONICS

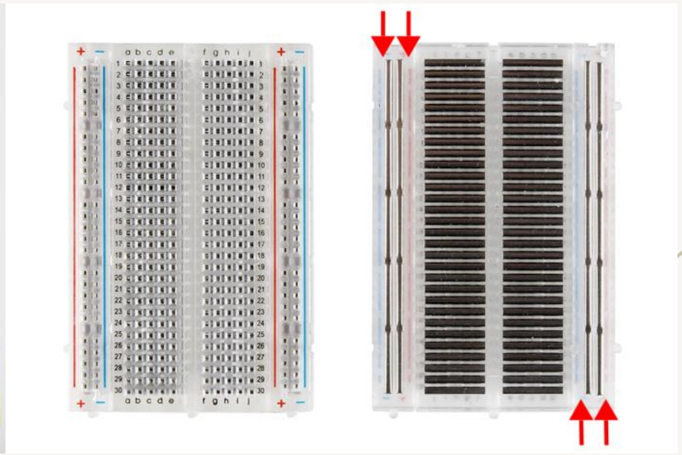
Fundamentals

- Power source (AC/DC)
- Voltage = Current x Resistance
- Ground, Ground, Ground
- If in doubt, there’s no doubt, double check & ask

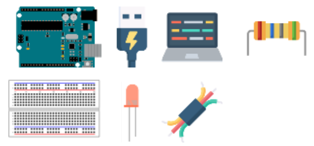
Breadboard?

- Build prototypes
- Connect using “hookup wire”





Hardware



Overview

☐ Connect External Hardware

☐ Modify SOS

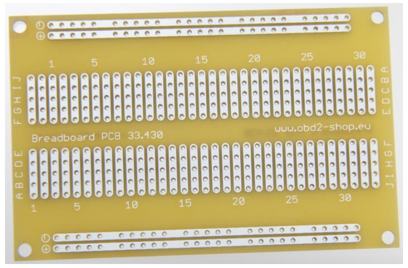
☐ Send command via Serial

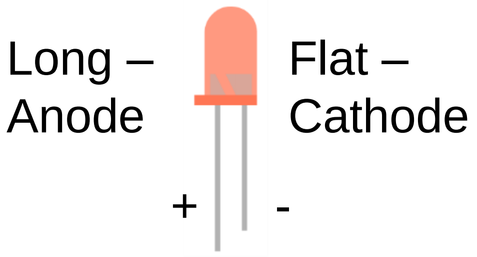
ACTIVITY 3 LIGHTBRINGER

Task: Blink an external LED to morse code (SOS) on command
Outcome

- Schematic reading (basic)
- Hardware + Software familiarization
- Serial & Serial Monitor

ACTIVITY 3 LIGHTBRINGER





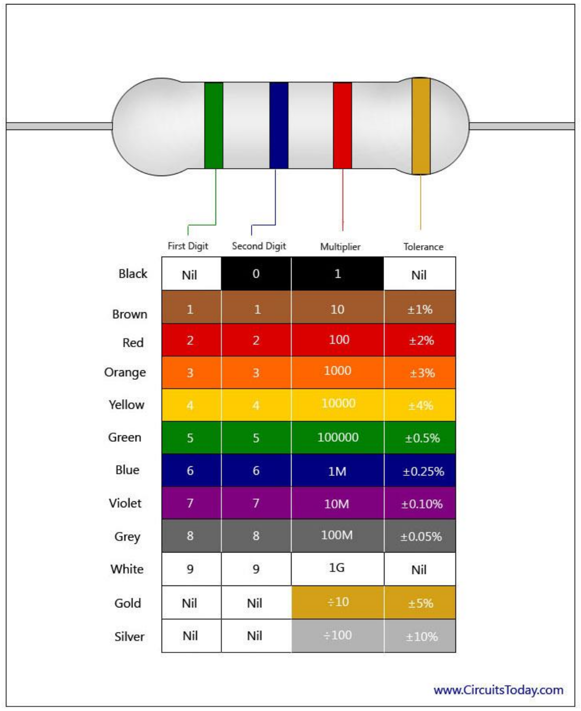
Hardware

Overview

- ☐ Connect External Hardware
- ☐ Modify SOS
- ☐ Send command via Serial

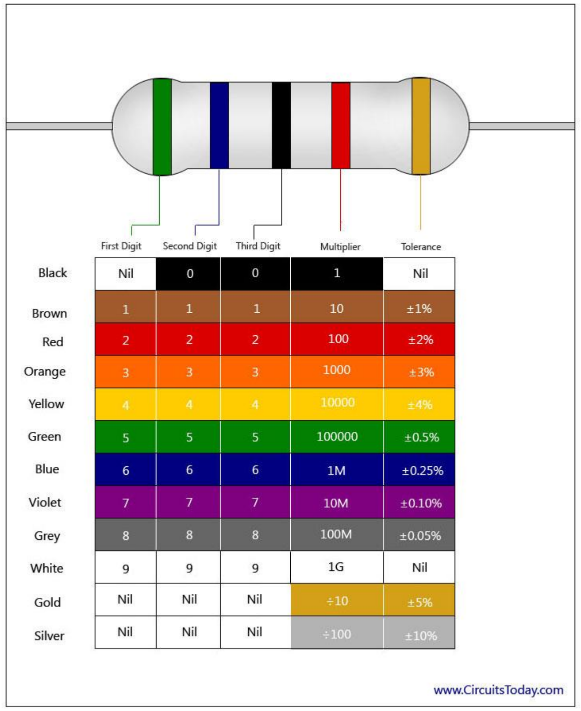
1. DISCONNECT!

4 Band Resistor


www.CircuitsToday.com

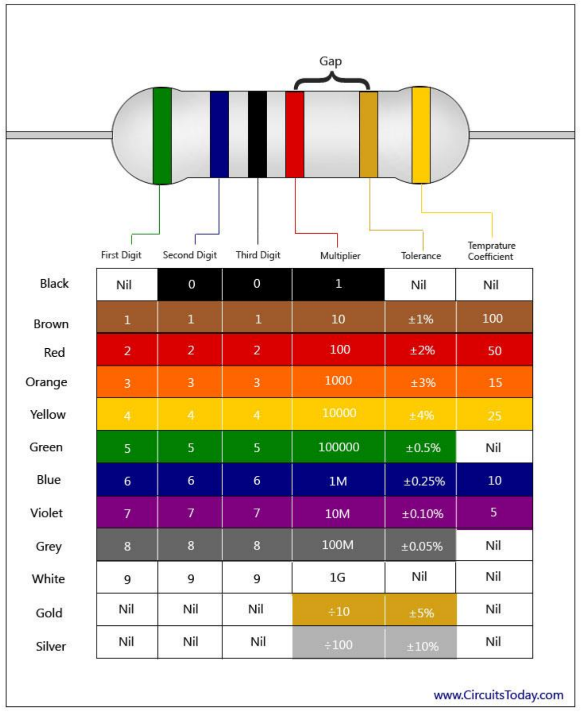
	First Digit	Second Digit	Multiplier	Tolerance
Black	Nil	0	1	Nil
Brown	1	1	10	±1%
Red	2	2	100	±2%
Orange	3	3	1000	±3%
Yellow	4	4	10000	±4%
Green	5	5	100000	±0.5%
Blue	6	6	1M	±0.25%
Violet	7	7	10M	±0.10%
Grey	8	8	100M	±0.05%
White	9	9	1G	Nil
Gold	Nil	Nil	÷10	±5%
Silver	Nil	Nil	÷100	±10%

5 Band Resistor


www.CircuitsToday.com

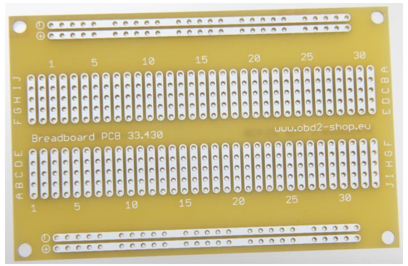
	First Digit	Second Digit	Third Digit	Multiplier	Tolerance
Black	Nil	0	0	1	Nil
Brown	1	1	1	10	±1%
Red	2	2	2	100	±2%
Orange	3	3	3	1000	±3%
Yellow	4	4	4	10000	±4%
Green	5	5	5	100000	±0.5%
Blue	6	6	6	1M	±0.25%
Violet	7	7	7	10M	±0.10%
Grey	8	8	8	100M	±0.05%
White	9	9	9	1G	Nil
Gold	Nil	Nil	Nil	÷10	±5%
Silver	Nil	Nil	Nil	÷100	±10%

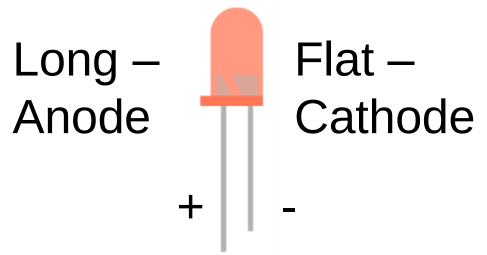
6 Band Resistor


Gap
www.CircuitsToday.com

	First Digit	Second Digit	Third Digit	Multiplier	Tolerance	Temprature Coefficient
Black	Nil	0	0	1	Nil	Nil
Brown	1	1	1	10	±1%	100
Red	2	2	2	100	±2%	50
Orange	3	3	3	1000	±3%	15
Yellow	4	4	4	10000	±4%	25
Green	5	5	5	100000	±0.5%	Nil
Blue	6	6	6	1M	±0.25%	10
Violet	7	7	7	10M	±0.10%	5
Grey	8	8	8	100M	±0.05%	Nil
White	9	9	9	1G	Nil	Nil
Gold	Nil	Nil	Nil	÷10	±5%	Nil
Silver	Nil	Nil	Nil	÷100	±10%	Nil

ACTIVITY 3
LIGHTBRINGER



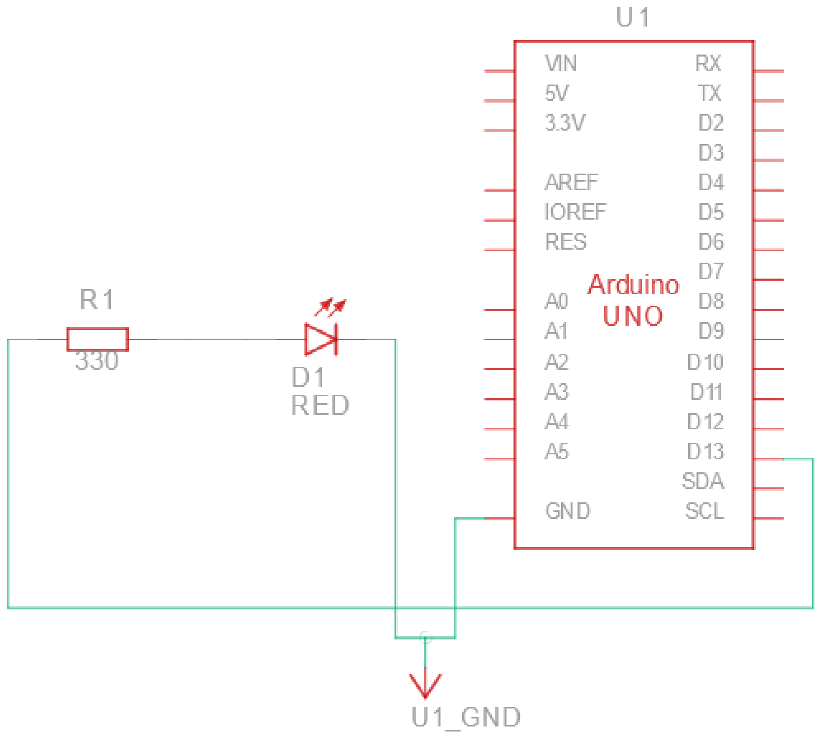


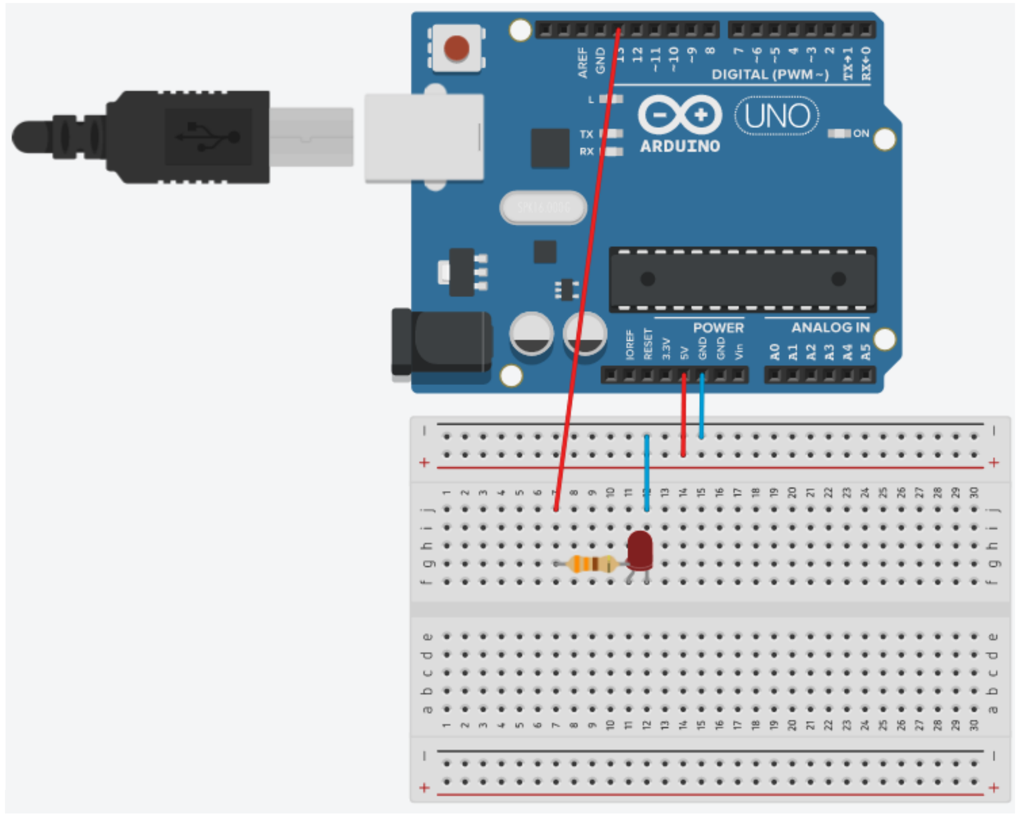
Hardware

Overview

- ☐ Connect External Hardware
- ☐ Modify SOS
- ☐ Send command via Serial

1. Wire as per schematic





ACTIVITY 3
LIGHTBRINGER

Hardware

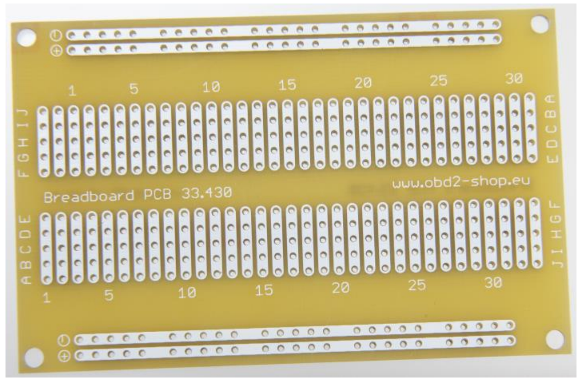
Overview

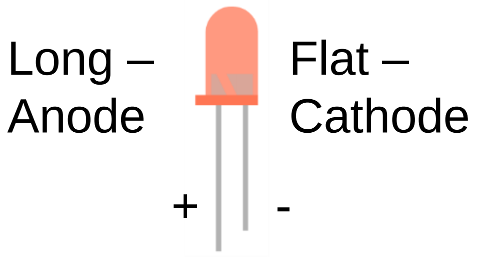
☐ Connect External Hardware

☐ Modify SOS

☐ Send command via Serial

Wiring Check





ACTIVITY 3
LIGHTBRINGER

Hardware

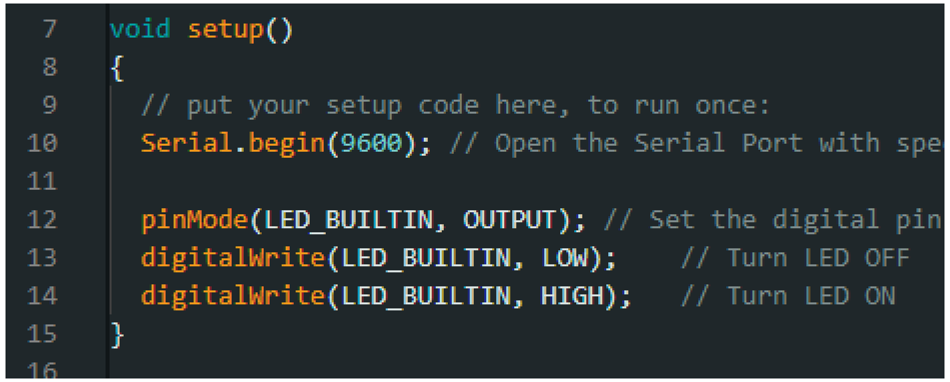
Overview

☑ Connect External Hardware

☐ Modify SOS

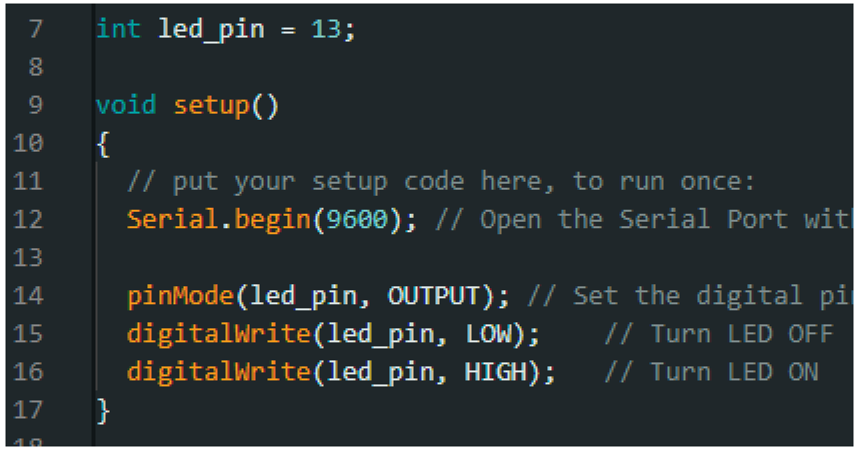
☐ Send command via Serial

1. Modify Activity 2 code (Blink for Help, SOS) to work with digital pin 13
2. Make a variable "led_pin" and swap it for "LED_BUILTIN" (everywhere, not only `setup()`)
3. Connect & Flash!



```
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with spe

  pinMode(LED_BUILTIN, OUTPUT); // Set the digital pin
  digitalWrite(LED_BUILTIN, LOW);    // Turn LED OFF
  digitalWrite(LED_BUILTIN, HIGH);   // Turn LED ON
}
```

$\Rightarrow$



```
int led_pin = 13;

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port wit

  pinMode(led_pin, OUTPUT); // Set the digital pi
  digitalWrite(led_pin, LOW);    // Turn LED OFF
  digitalWrite(led_pin, HIGH);   // Turn LED ON
}
```

ACTIVITY 3
LIGHTBRINGER

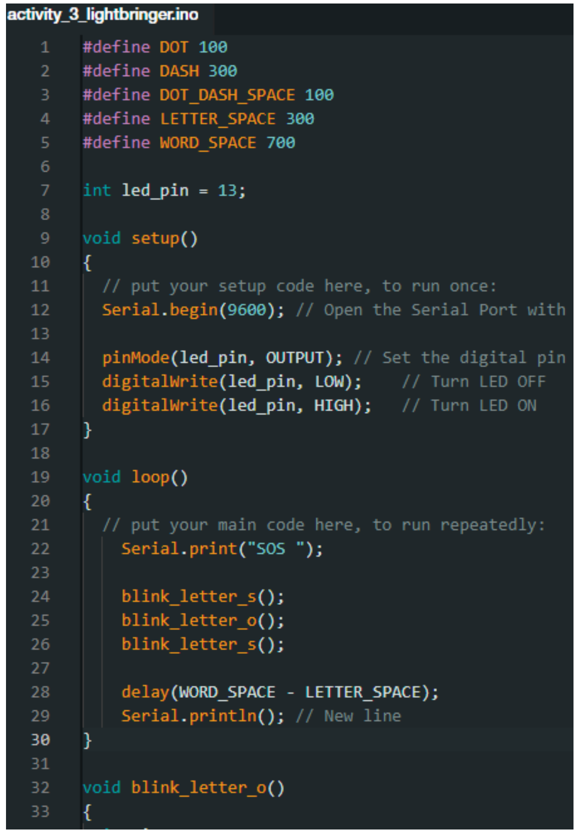
Hardware

Overview

☑ Connect External Hardware

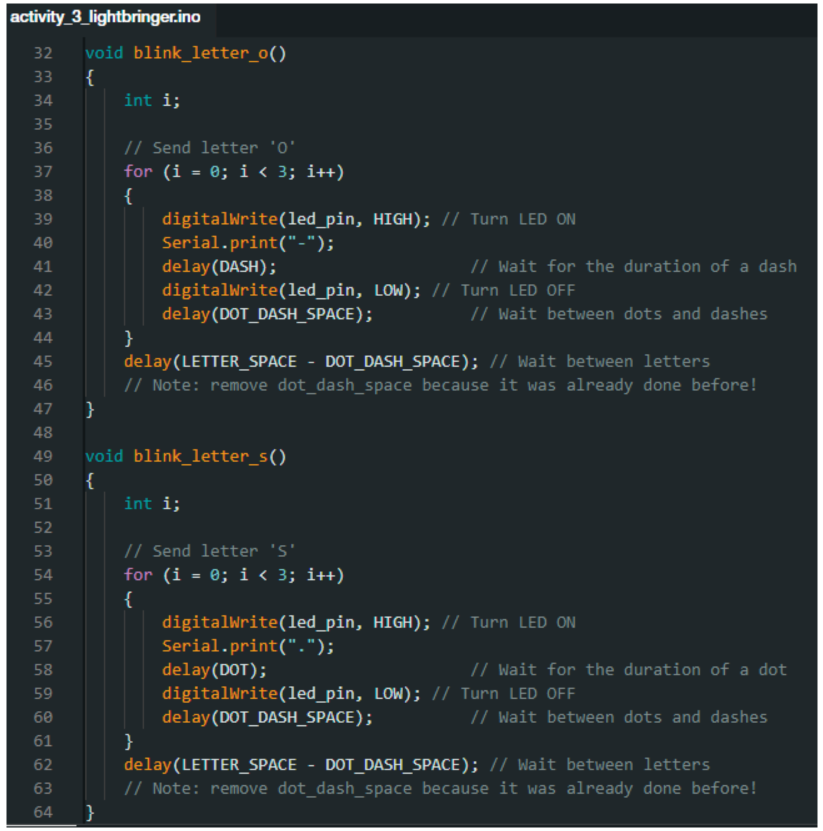
☐ **Modify SOS**

☐ Send command via Serial



activity_3_lightbringer.ino

```
#define DOT 100
#define DASH 300
#define DOT_DASH_SPACE 100
#define LETTER_SPACE 300
#define WORD_SPACE 700

int led_pin = 13;

void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600); // Open the Serial Port with

  pinMode(led_pin, OUTPUT); // Set the digital pin
  digitalWrite(led_pin, LOW);    // Turn LED OFF
  digitalWrite(led_pin, HIGH);   // Turn LED ON
}

void loop()
{
  // put your main code here, to run repeatedly:
    Serial.print("SOS ");

    blink_letter_s();
    blink_letter_o();
    blink_letter_s();

    delay(WORD_SPACE - LETTER_SPACE);
    Serial.println(); // New line
}

void blink_letter_o()
{
```



activity_3_lightbringer.ino

```
void blink_letter_o()
{
    int i;

    // Send letter 'O'
    for (i = 0; i < 3; i++)
    {
        digitalWrite(led_pin, HIGH); // Turn LED ON
        Serial.print("-");
        delay(DASH);                    // Wait for the duration of a dash
        digitalWrite(led_pin, LOW); // Turn LED OFF
        delay(DOT_DASH_SPACE);          // Wait between dots and dashes
    }
    delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
    // Note: remove dot_dash_space because it was already done before!
}

void blink_letter_s()
{
    int i;

    // Send letter 'S'
    for (i = 0; i < 3; i++)
    {
        digitalWrite(led_pin, HIGH); // Turn LED ON
        Serial.print(".");
        delay(DOT);                     // Wait for the duration of a dot
        digitalWrite(led_pin, LOW); // Turn LED OFF
        delay(DOT_DASH_SPACE);          // Wait between dots and dashes
    }
    delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
    // Note: remove dot_dash_space because it was already done before!
}
```

ACTIVITY 3
LIGHTBRINGER

Hardware

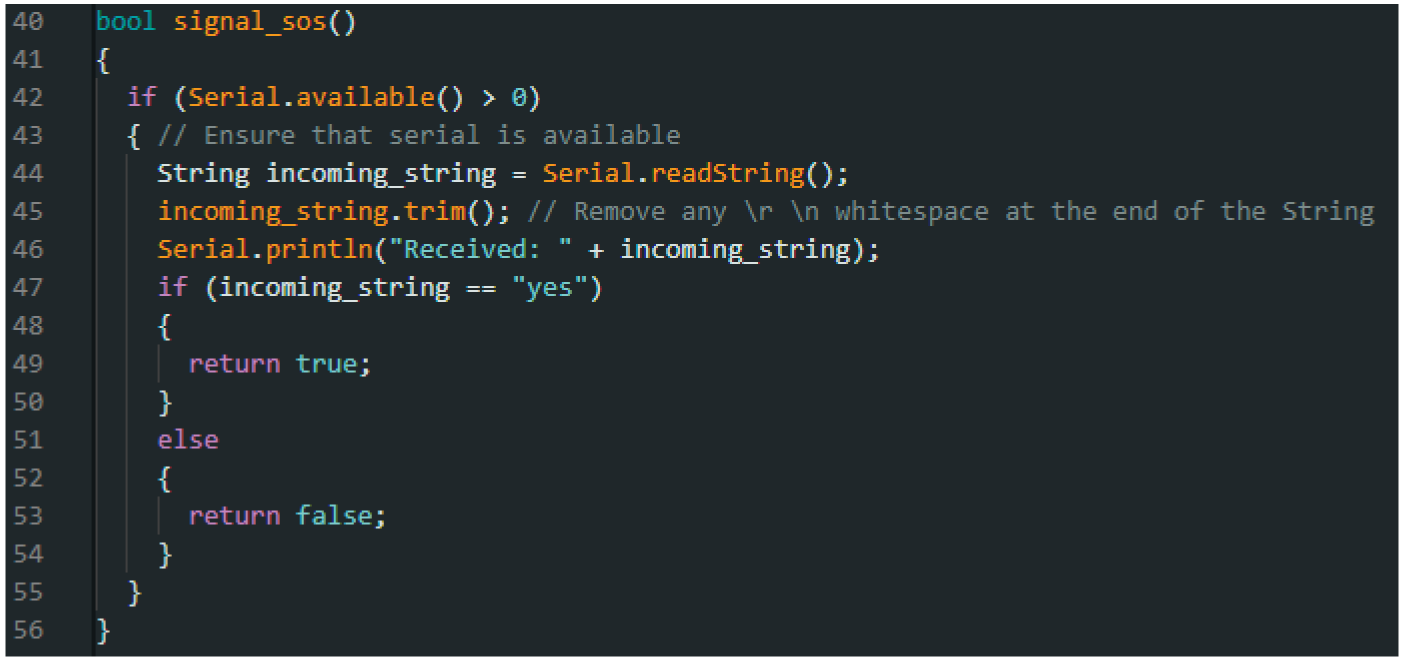
Overview

☑ Connect External Hardware

☑ Modify SOS

☐ Send command via Serial

1. Serial Monitor is not a pure monitor!
2. Use `Serial.readString()`



```
bool signal_sos()
{
  if (Serial.available() > 0)
  { // Ensure that serial is available
    String incoming_string = Serial.readString();
    incoming_string.trim(); // Remove any \r \n whitespace at the end of the String
    Serial.println("Received: " + incoming_string);
    if (incoming_string == "yes")
    {
      return true;
    }
    else
    {
      return false;
    }
  }
}
```

ACTIVITY 3
LIGHTBRINGER

Hardware

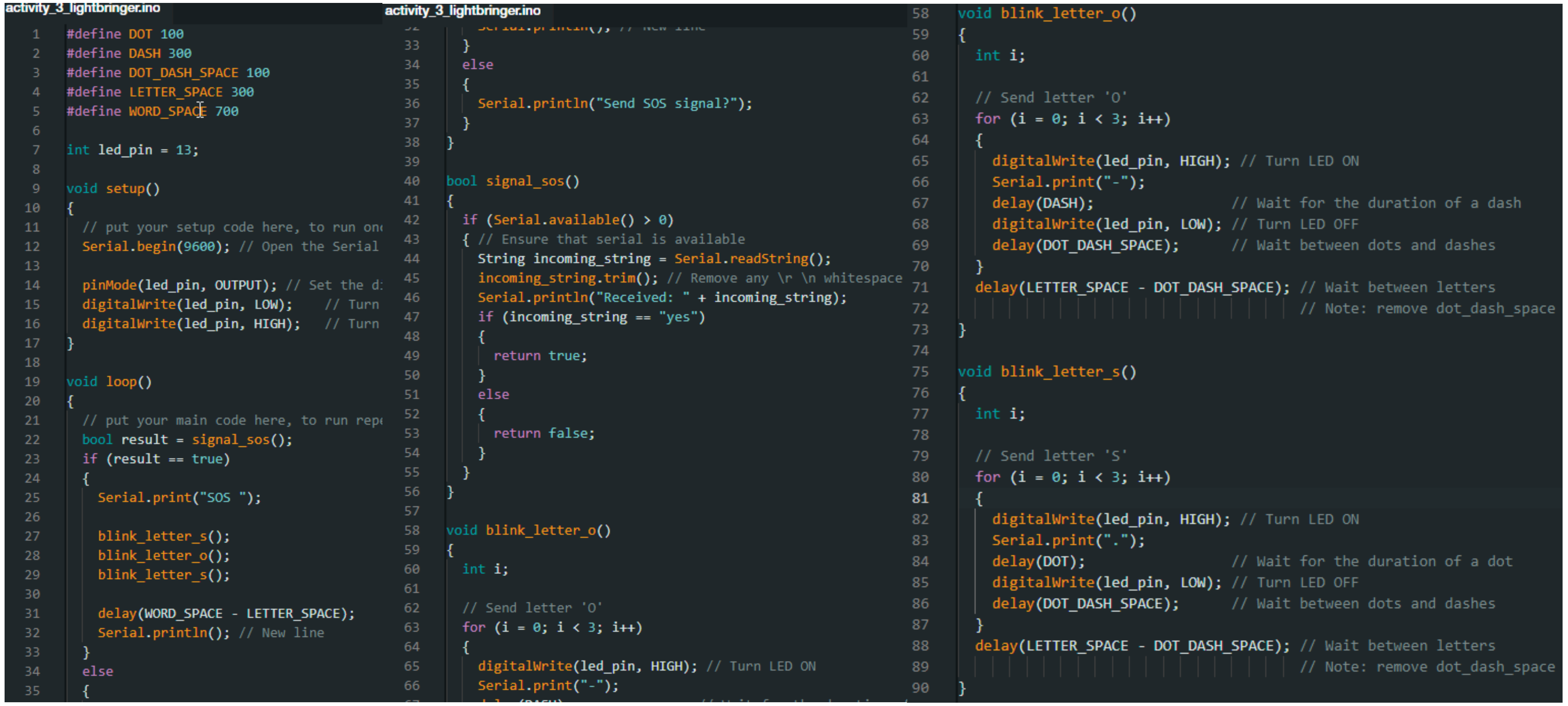
Overview

☑ Connect External Hardware

☑ Modify SOS

☐ Send command via Serial

1. Something like this? Flash!



activity_3_lightbringer.ino

```
#define DOT 100
#define DASH 300
#define DOT_DASH_SPACE 100
#define LETTER_SPACE 300
#define WORD_SPACE 700

int led_pin = 13;

void setup()
{
  // put your setup code here, to run onc
  Serial.begin(9600); // Open the Serial

  pinMode(led_pin, OUTPUT); // Set the d:
  digitalWrite(led_pin, LOW);     // Turn
  digitalWrite(led_pin, HIGH);    // Turn
}

void loop()
{
  // put your main code here, to run repe
  bool result = signal_sos();
  if (result == true)
  {
    Serial.print("SOS ");

    blink_letter_s();
    blink_letter_o();
    blink_letter_s();

    delay(WORD_SPACE - LETTER_SPACE);
    Serial.println(); // New line
  }
  else
  {
```

activity_3_lightbringer.ino

```
  }
  else
  {
    Serial.println("Send SOS signal?");
  }
}

bool signal_sos()
{
  if (Serial.available() > 0)
  { // Ensure that serial is available
    String incoming_string = Serial.readString();
    incoming_string.trim(); // Remove any \r \n whitespace
    Serial.println("Received: " + incoming_string);
    if (incoming_string == "yes")
    {
      return true;
    }
    else
    {
      return false;
    }
  }
}

void blink_letter_o()
{
  int i;

  // Send letter 'O'
  for (i = 0; i < 3; i++)
  {
    digitalWrite(led_pin, HIGH); // Turn LED ON
    Serial.print("-");
```

```
void blink_letter_o()
{
  int i;

  // Send letter 'O'
  for (i = 0; i < 3; i++)
  {
    digitalWrite(led_pin, HIGH); // Turn LED ON
    Serial.print("-");
    delay(DASH);                 // Wait for the duration of a dash
    digitalWrite(led_pin, LOW); // Turn LED OFF
    delay(DOT_DASH_SPACE);      // Wait between dots and dashes
  }
  delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
                                        // Note: remove dot_dash_space
}

void blink_letter_s()
{
  int i;

  // Send letter 'S'
  for (i = 0; i < 3; i++)
  {
    digitalWrite(led_pin, HIGH); // Turn LED ON
    Serial.print(".");
    delay(DOT);                  // Wait for the duration of a dot
    digitalWrite(led_pin, LOW); // Turn LED OFF
    delay(DOT_DASH_SPACE);      // Wait between dots and dashes
  }
  delay(LETTER_SPACE - DOT_DASH_SPACE); // Wait between letters
                                        // Note: remove dot_dash_space
}
```

ACTIVITY 3
LIGHTBRINGER

Hardware

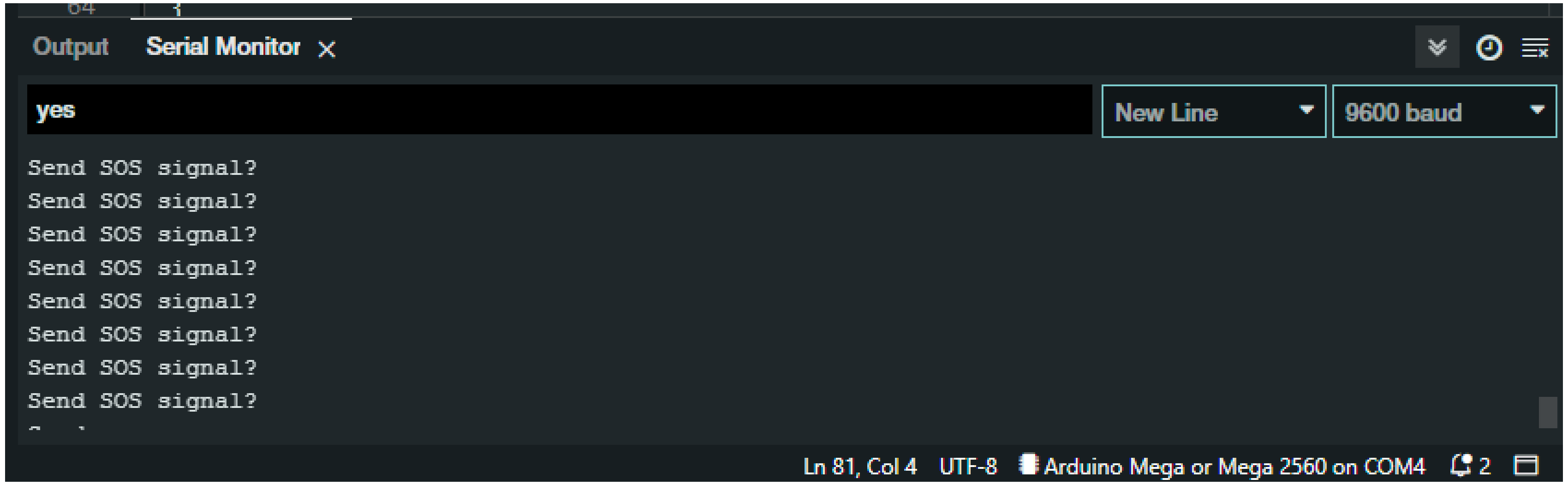
Overview

☑ Connect External Hardware

☑ Modify SOS

☐ Send command via Serial

1. "Ctrl + Shift + M" to open Serial Monitor
2. "Ctrl + Enter" to send message



```
yes
```

New Line | 9600 baud

```
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
```

Ln 81, Col 4   UTF-8   Arduino Mega or Mega 2560 on COM4   2

ACTIVITY 3
LIGHTBRINGER

Hardware

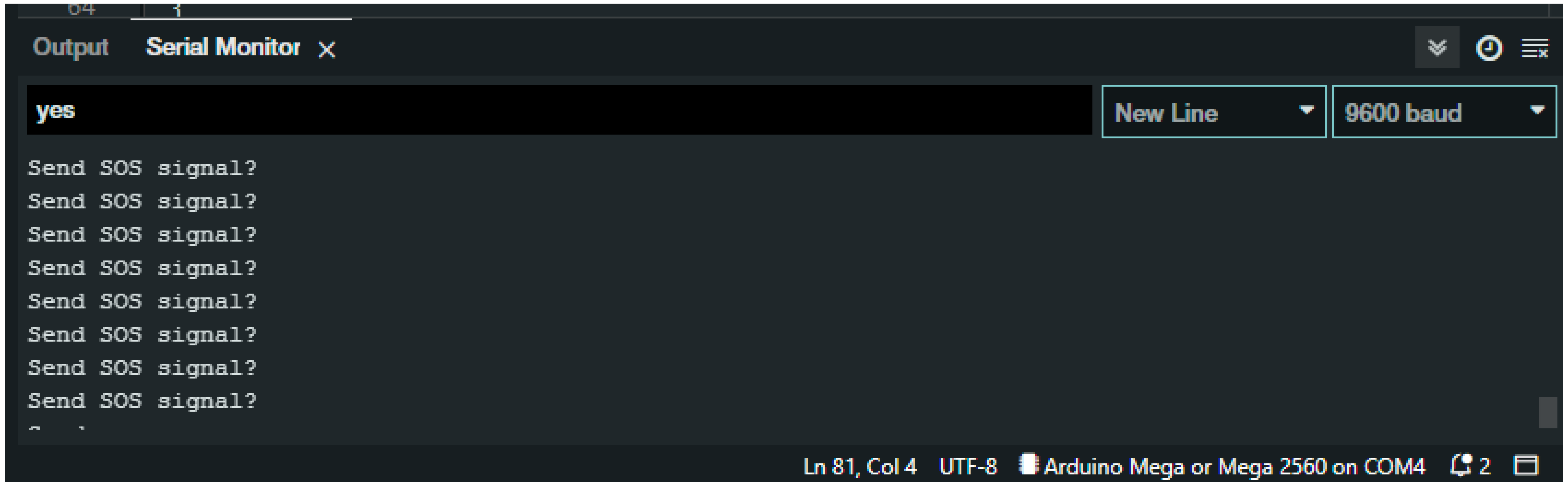
Overview

☑ Connect External Hardware

☑ Modify SOS

☐ Send command via Serial

1. Note the option to change line endings.
2. How quickly is your command registered?
3. What happens if you change the "baud" rate?



```
yes
```

```
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
Send SOS signal?
```

Hardware

Overview

☑ Connect External Hardware

☑ Modify SOS

☑ Send command via Serial

ACTIVITY 3
LIGHTBRINGER

Task: Blink an external LED to morse code (SOS) on command

Outcome

- Schematic reading (basic)
- Hardware + Software familiarization
- Serial & Serial Monitor

Recap

- Build your own circuits
- Knowledge in documentation and/or datasheets!
- Know the limitations (software and hardware)

Hardware

Overview

☐ More LEDs

☐ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

ACTIVITY 4
KNIGHT RIDER

Task: Blink LEDs from left to right. Change speed with buttons.
Outcome

- Hardware + Software familiarization
- Input / Sensor

BREAK TIME
RESUME AT:

Hardware

Overview

☐ More LEDs

☐ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

ACTIVITY 4
KNIGHT RIDER

Task: Blink LEDs from left to right. Change speed with buttons.
Outcome

- Hardware + Software familiarization
- Input / Sensor

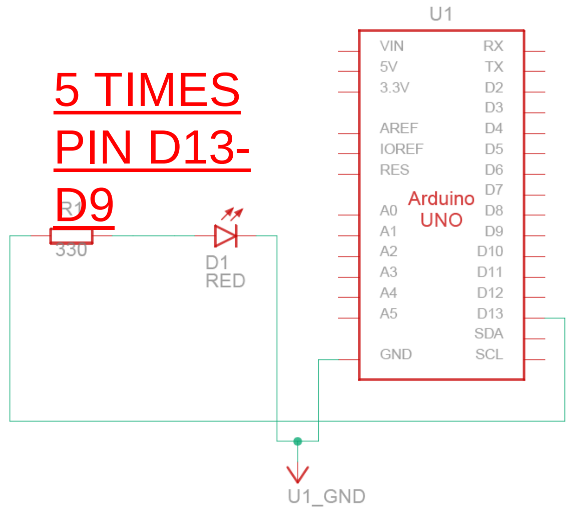
ACTIVITY 4
KNIGHT RIDER

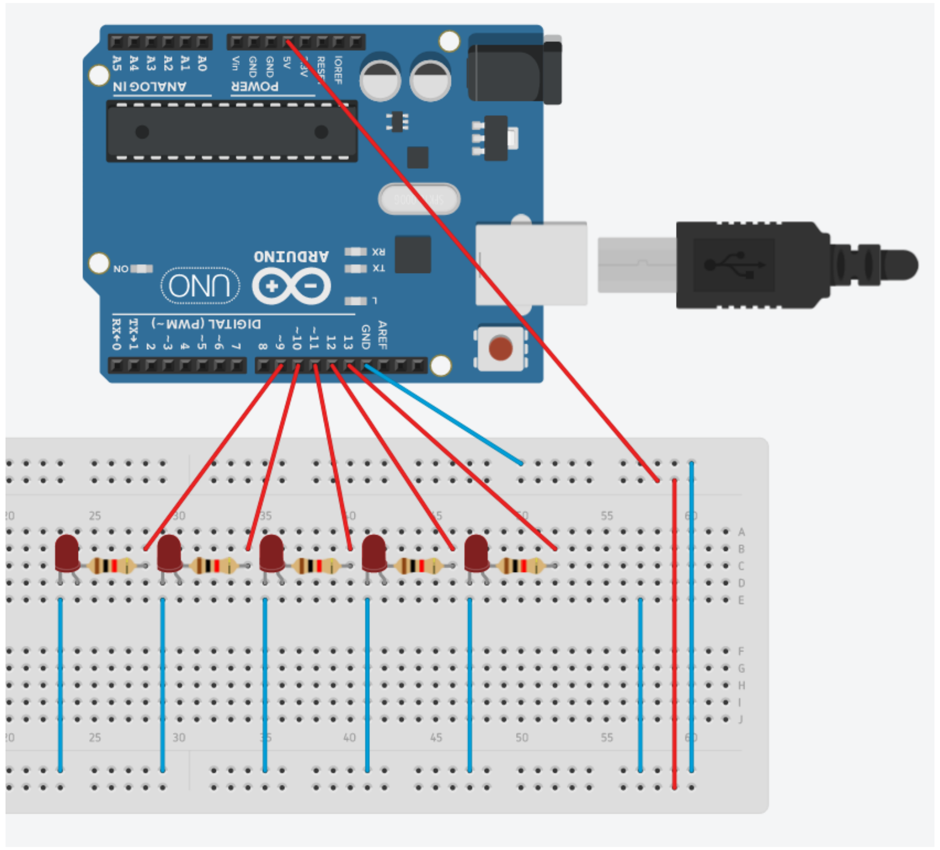
Hardware

Overview

- ☐ More LEDs
- ☐ Buttons (Pull-up config) & Debouncing
- ☐ Code "Knight Rider" effect

1. Disconnect
2. Wire as per schematic





ACTIVITY 4
KNIGHT RIDER

Hardware

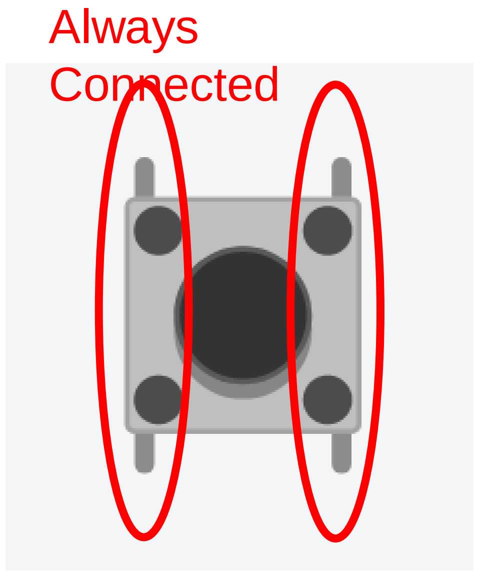
Overview

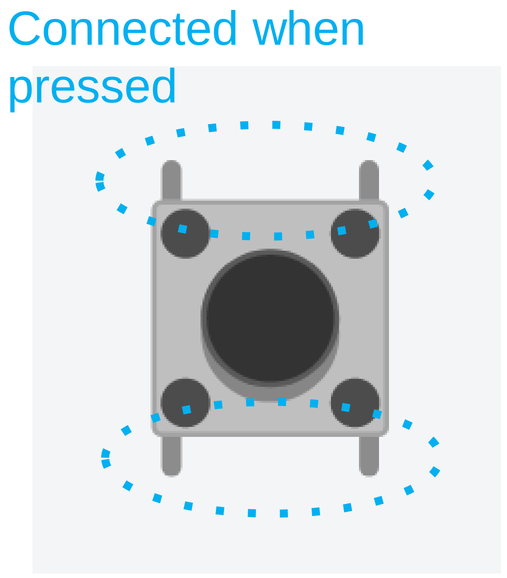
☑ More LEDs

☐ Buttons (Pull-up config) & Debouncing

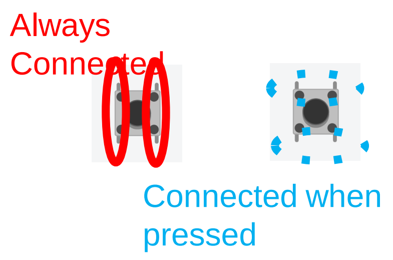
☐ Code “Knight Rider” effect

1. Push button





ACTIVITY 4
KNIGHT RIDER



Hardware

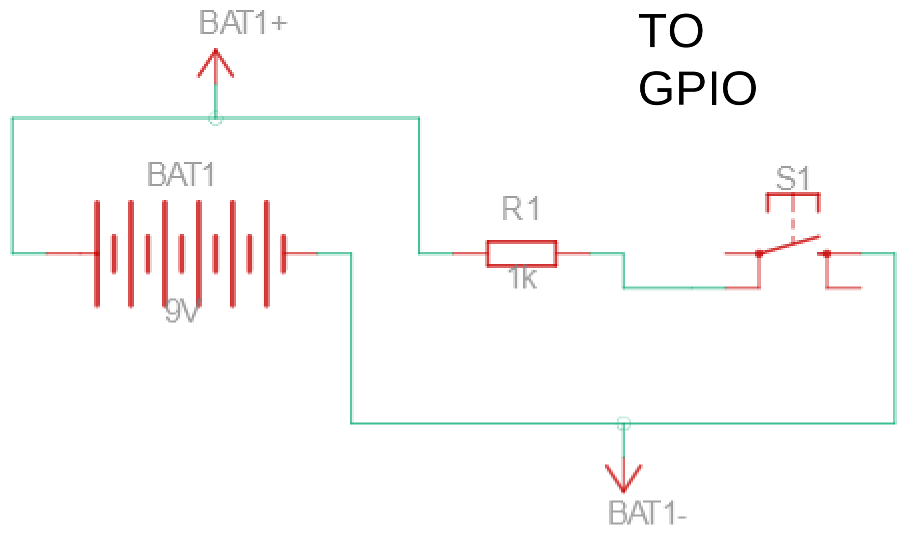
Overview

☑ More LEDs

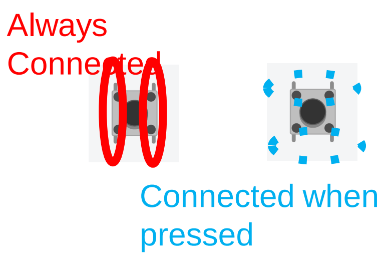
☐ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

1. Pull-up configuration
2. Default value is HIGH
3. Pressed value is LOW



ACTIVITY 4
KNIGHT RIDER



Hardware

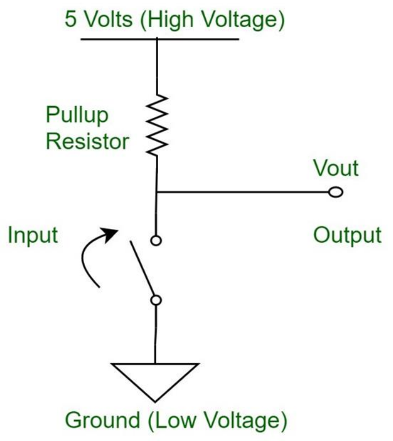
Overview

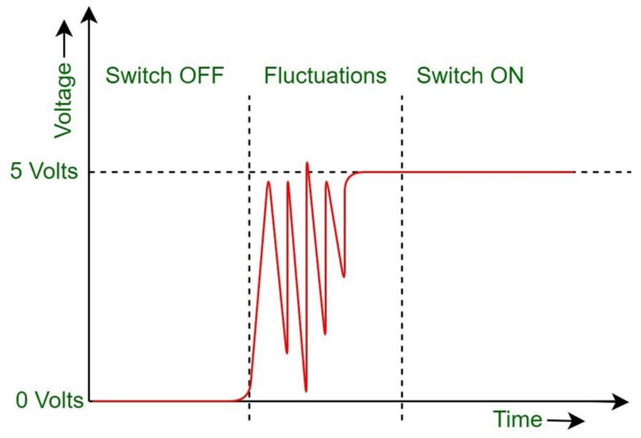
☑ More LEDs

☐ Buttons (Pull-up config) & Debouncing

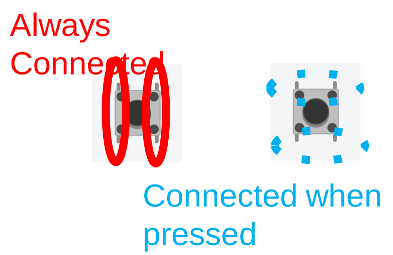
☐ Code "Knight Rider" effect

1. Things are not perfect! (;_;)
2. Unwanted fluctuations (Aka noise)
3. Solution: add a capacitor to smoothen out noise





ACTIVITY 4
KNIGHT RIDER



Hardware

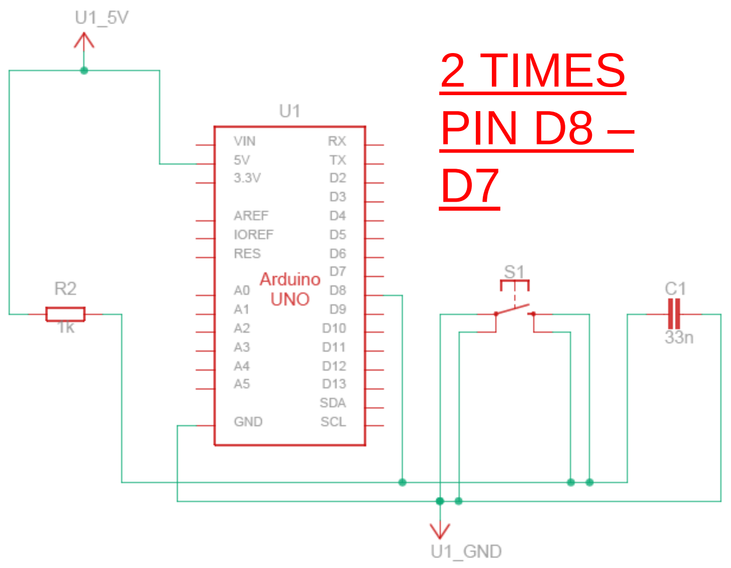
Overview

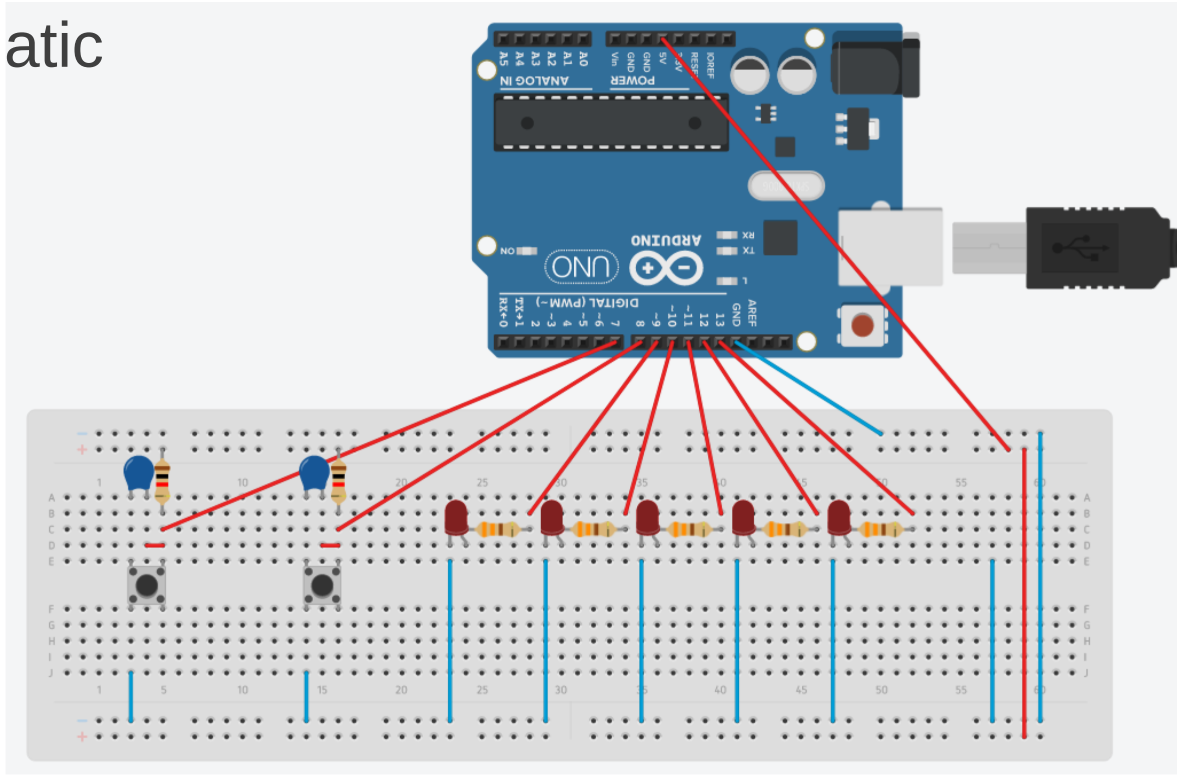
☑ More LEDs

☐ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

1. Wire as per Schematic





ACTIVITY 4
KNIGHT RIDER

Hardware

Overview

☑ More LEDs

☐ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

Wiring Check

ACTIVITY 4
KNIGHT RIDER

Hardware

Overview

☑ More LEDs

☑ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

1. Try coding a knight rider effect (single direction)

Pseudocode:

```
Int delay_time (has max and min values)
setup:
    led pins as output
    pushbtn pins as input
main:
    if push btn1 only: increment delay_time
    if push btn2 only: decrement delay_time
    turn LED 1 ON, wait delay_time, turn LED 1 OFF
    …
    turn LED 5 ON, wait delay_time, turn LED 5 OFF
```

5ms <= time <= 150ms
default value = 100ms
increment = 5ms

Get input with `digitalRead(pin)`

ACTIVITY 4
KNIGHT RIDER

Hardware

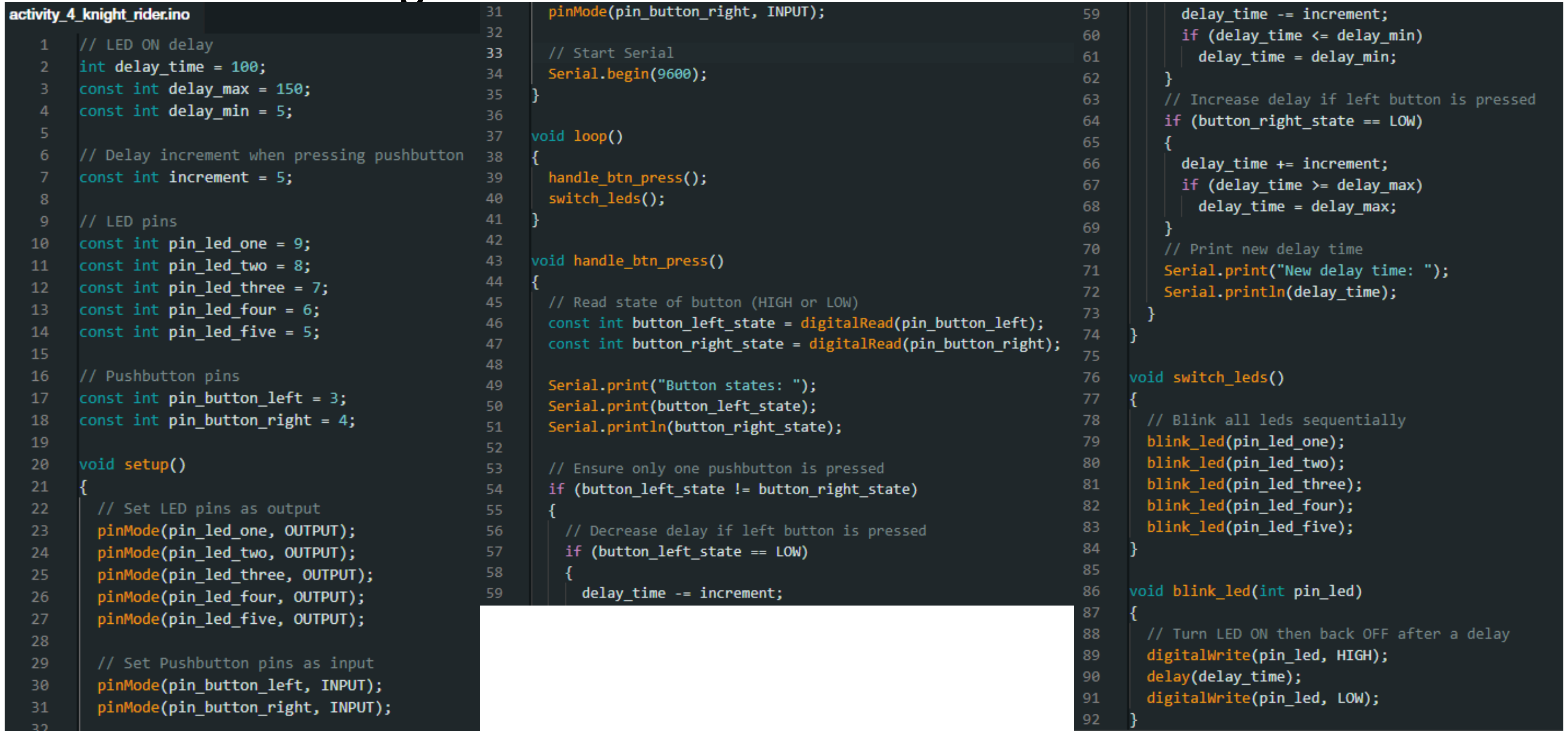
Overview

☑ More LEDs

☑ Buttons (Pull-up config) & Debouncing

☐ Code “Knight Rider” effect

1. Something like this? Flash!



activity_4_knight_rider.ino

```
// LED ON delay
int delay_time = 100;
const int delay_max = 150;
const int delay_min = 5;

// Delay increment when pressing pushbutton
const int increment = 5;

// LED pins
const int pin_led_one = 9;
const int pin_led_two = 8;
const int pin_led_three = 7;
const int pin_led_four = 6;
const int pin_led_five = 5;

// Pushbutton pins
const int pin_button_left = 3;
const int pin_button_right = 4;

void setup()
{
  // Set LED pins as output
  pinMode(pin_led_one, OUTPUT);
  pinMode(pin_led_two, OUTPUT);
  pinMode(pin_led_three, OUTPUT);
  pinMode(pin_led_four, OUTPUT);
  pinMode(pin_led_five, OUTPUT);

  // Set Pushbutton pins as input
  pinMode(pin_button_left, INPUT);
  pinMode(pin_button_right, INPUT);
```

```
  pinMode(pin_button_right, INPUT);

  // Start Serial
  Serial.begin(9600);
}

void loop()
{
  handle_btn_press();
  switch_leds();
}

void handle_btn_press()
{
  // Read state of button (HIGH or LOW)
  const int button_left_state = digitalRead(pin_button_left);
  const int button_right_state = digitalRead(pin_button_right);

  Serial.print("Button states: ");
  Serial.print(button_left_state);
  Serial.println(button_right_state);

  // Ensure only one pushbutton is pressed
  if (button_left_state != button_right_state)
  {
    // Decrease delay if left button is pressed
    if (button_left_state == LOW)
    {
      delay_time -= increment;
```

```
      delay_time -= increment;
      if (delay_time <= delay_min)
        delay_time = delay_min;
    }
    // Increase delay if left button is pressed
    if (button_right_state == LOW)
    {
      delay_time += increment;
      if (delay_time >= delay_max)
        delay_time = delay_max;
    }
    // Print new delay time
    Serial.print("New delay time: ");
    Serial.println(delay_time);
  }
}

void switch_leds()
{
  // Blink all leds sequentially
  blink_led(pin_led_one);
  blink_led(pin_led_two);
  blink_led(pin_led_three);
  blink_led(pin_led_four);
  blink_led(pin_led_five);
}

void blink_led(int pin_led)
{
  // Turn LED ON then back OFF after a delay
  digitalWrite(pin_led, HIGH);
  delay(delay_time);
  digitalWrite(pin_led, LOW);
}
```

Hardware

Overview

☑ More LEDs

☑ Buttons (Pull-up config) & Debouncing

☑ Code “Knight Rider” effect

ACTIVITY 4
KNIGHT RIDER

Task: Blink LEDs from left to right. Change speed with buttons.
Outcome

- Hardware + Software familiarization
- Input / Sensor

Recap

- Input / Sensor are used to gather data from the world
- Real life has “defects” compared to models

QUESTIONS?

DEMO 1
LED/SERVO CONTROLLER

Demo	Example Application
Battery (power bank)	Wildlife tracking, Autonomous robot
Remote control (ESP8266)	Smart TV, Robotic surgery, Marine exploration
E-paper screen	Name tag, Price tag, Bus stop ads

DEMO 2
RASPBERRY PI

1. Raspbian on raspberry pi
2. Retro game arcade
3. Web server

WHAT NEXT?

Buy your own?

Polling vs Interrupts

ESP8266? Zigbee? Bluetooth? WiFi?

IoT benefits and dangers

Slides, files, source code available:

https://github.com/MrTanoshii/Intro-Embedded-Systems

QUESTIONS?

THANK YOU

Tsen Chee Vincent LEUNG YIN KO

vincent.leungyinko@gmail.com

Slides, files, source code available:

https://github.com/MrTanoshii/Intro-Embedded-Systems