Co-funded by the
Erasmus+ Programme
of the European Union

Trainer Handouts

#	Module	#	Handout
A	Microcontrollers	A1	Basic introduction to Microcontrollers
		A2	Logic Gates
		A3	Introduction to (Block) coding
		A4	Arduino Basics
		A5	Introduction to IoT
B	3D Objects	B1	3D Modelling
		B2	3D Printing
		B3	Photogrammetry
C	App Development	C1	Introduction to MIT App Inventor 2
D	Web Development	D1	HTML/CSS/ JavaScript
		D2	Wordpress

DigComp 2.0 - the Conceptual Reference Model

In this section, you can find the reference model for the Digital Competence Framework for Citizens (published by the EU), which are mentioned in the Trainer Handouts.

Competence areas Dimension 1	**Competences** Dimension 2
1. Information and data literacy	**1.1 Browsing, searching and filtering data, information and digital content** To articulate information needs, to search for data, information and content in digital environments, to access them and to navigate between them. To create and update personal search strategies. **1.2 Evaluating data, information and digital content** To analyse, compare and critically evaluate the credibility and reliability of sources of data, information and digital content. To analyse, interpret and critically evaluate the data, information and digital content. **1.3 Managing data, information and digital content** To organise, store and retrieve data, information and content in digital environments. To organise and process them in a structured environment.
2. Communication and collaboration	**2.1 Interacting through digital technologies** To interact through a variety of digital technologies and to understand appropriate digital communication means for a given context. **2.2 Sharing through digital technologies** To share data, information and digital content with others through appropriate digital technologies. To act as an intermediary, to know about referencing and attribution practices. **2.3 Engaging in citizenship through digital technologies** To participate in society through the use of public and private digital services. To seek opportunities for self-empowerment and for participatory citizenship through appropriate digital technologies. **2.4 Collaborating through digital technologies** To use digital tools and technologies for collaborative processes, and for co-construction and co-creation of resources and knowledge. **2.5 Netiquette** To be aware of behavioural norms and know-how while using digital technologies and interacting in digital environments. To adapt communication strategies to the specific audience and to be aware of cultural and generational diversity in digital environments. **2.6 Managing digital identity** To create and manage one or multiple digital identities, to be able to protect one's own reputation, to deal with the data that one produces through several digital tools, environments and services.

3. Digital content creation	**3.1 Developing digital content** To create and edit digital content in different formats, to express oneself through digital means. **3.2 Integrating and re-elaborating digital content** To modify, refine, improve and integrate information and content into an existing body of knowledge to create new, original and relevant content and knowledge. **3.3 Copyright and licences** To understand how copyright and licences apply to data, information and digital content. **3.4 Programming** To plan and develop a sequence of understandable instructions for a computing system to solve a given problem or perform a specific task.
4. Safety	**4.1 Protecting devices** To protect devices and digital content, and to understand risks and threats in digital environments. To know about safety and security measures and to have due regard to reliability and privacy. **4.2 Protecting personal data and privacy** To protect personal data and privacy in digital environments. To understand how to use and share personally identifiable information while being able to protect oneself and others from damages. To understand that digital services use a “Privacy policy” to inform how personal data is used. **4.3 Protecting health and well-being** To be able to avoid health-risks and threats to physical and psychological well-being while using digital technologies. To be able to protect oneself and others from possible dangers in digital environments (e.g. cyber bullying). To be aware of digital technologies for social wellbeing and social inclusion. **4.4 Protecting the environment** To be aware of the environmental impact of digital technologies and their use.
5. Problem solving	**5.1 Solving technical problems** To identify technical problems when operating devices and using digital environments, and to solve them (from trouble-shooting to solving more complex problems). **5.2 Identifying needs and technological responses** To assess needs and to identify, evaluate, select and use digital tools and possible technological responses to solve them. To adjust and customise digital environments to personal needs (e.g. accessibility). **5.3 Creatively using digital technologies** To use digital tools and technologies to create knowledge and to innovate processes and products. To engage individually and collectively in cognitive processing to understand and resolve conceptual problems and problem situations in digital environments. **5.4 Identifying digital competence gaps** To understand where one’s own digital competence needs to be improved or updated. To be able to support others with their digital competence development. To seek opportunities for self-development and to keep up-to-date with the digital evolution.

Module: Microcontroller

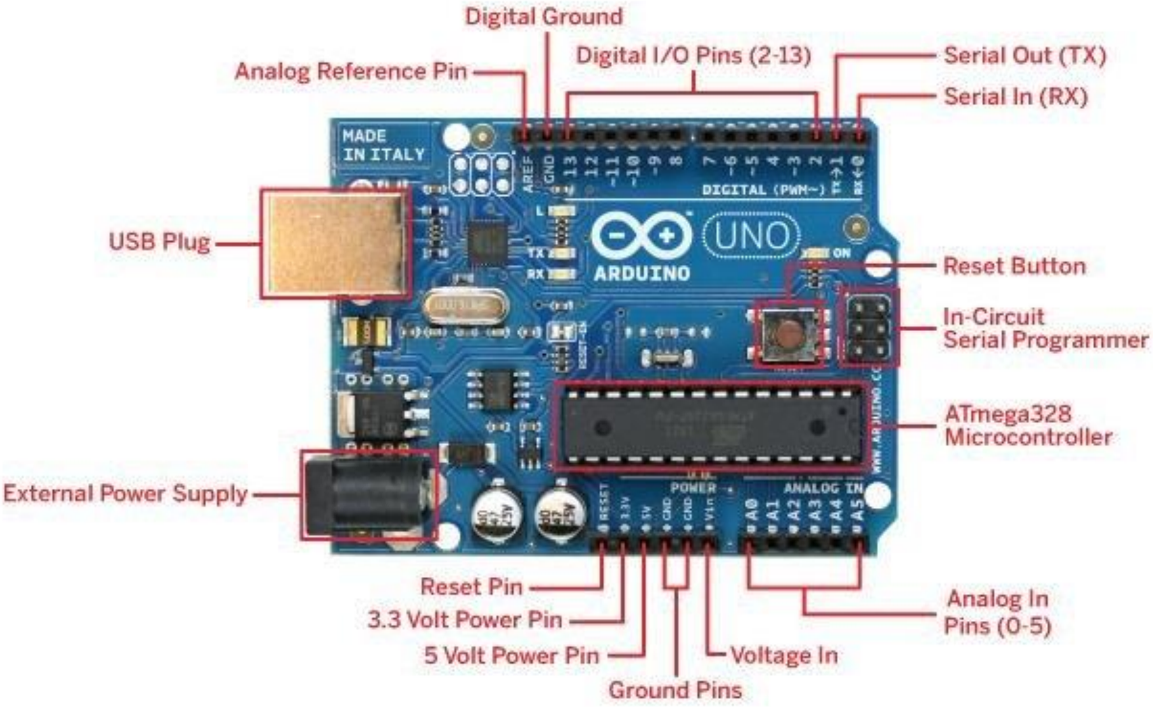
Topic: Introduction to microcontrollers

General Description:

The handout is intended to give an overview of the microcontrollers used in the learning offers. As for now these are Arduino, micro:bit and Raspberry Pi.
There will be no additional task sheets for this handout since it is only meant for the trainer. You can use everything in this handout for your lessons.

Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.



- **GND:** Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V & 3.3V:** The 5V pin supplies 5 volts of power and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog:** The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

- **Digital:** Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM:** You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF:** Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Additional link or information:

A guide with information about Arduino and also some basic projects you could start with to get used to Arduino: https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40

Micro:bit

What is a micro:bit?

It is a pocket-sized computer, 70 times smaller and 18 times faster than the original BBC Micro computers used in schools. It has 25 red LED lights that can flash messages and be used to create games.

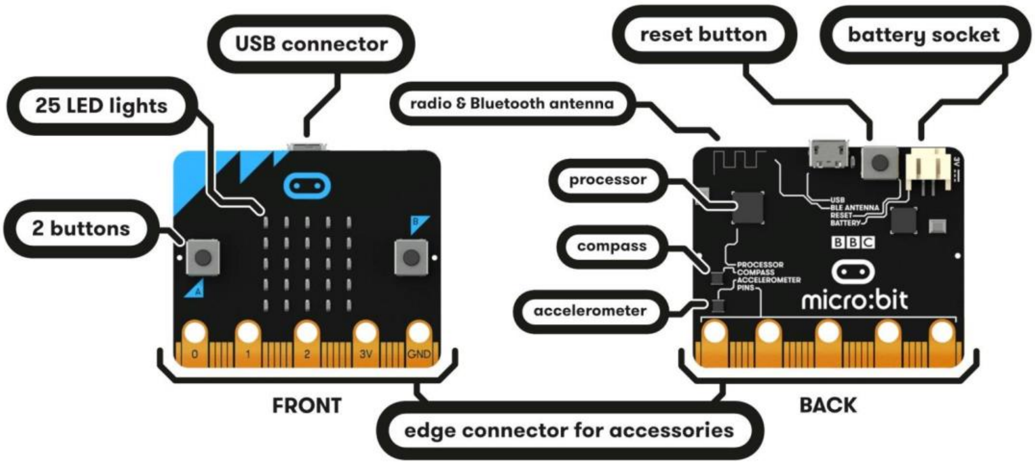
There are two programmable buttons that can be used to control games or pause and skip songs on a playlist.

It has an accelerometer so it can detect motion and knows when you are on the move. The built-in compass knows which direction you are heading in and it can use a low energy Bluetooth connection to interact with other devices and the Internet.

How do you use a micro bit?

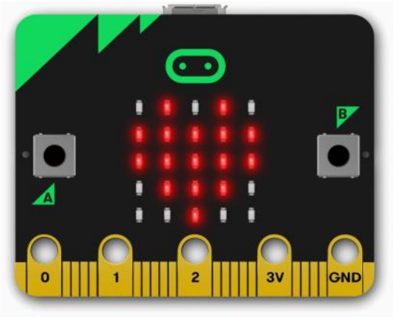
- Step 1: Connect your BBC micro:bit to your computer. Connect the small end of the USB cable to the micro USB port on your BBC micro:bit.
- Step 2: Download your program.
- Step 3: Flash the file to your BBC micro:bit.

What are the features of a micro bit?



- 25 individually-programmable LEDs
- 2 programmable buttons
- Physical connection pins
- Light and temperature sensors
- Motion sensors (accelerometer and compass)
- Wireless Communication, via Radio and Bluetooth
- USB-Interface

LEDs



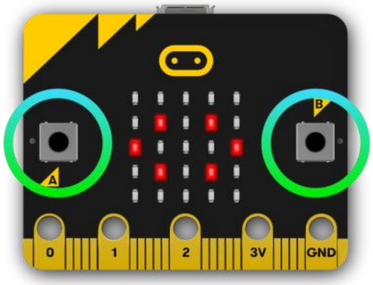
What is it? LED stands for Light Emitting Diode. The micro:bit has 25 individually-programmable LEDs, allowing you to display text, numbers, and images.

How do I code it? Learn more about coding the LEDs, or use the code references below.

Python or MakeCode

Examples - check out this Animated Flashing Heart in JavaScript or learn how to make animations with Python!

Buttons



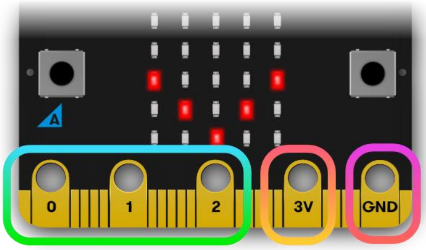
What is it? There are two buttons on the front of the micro:bit (labelled A and B). You can detect when these buttons are pressed, allowing you to trigger code on the device.

How do I code it? See the code references below.

Python or MakeCode

Examples - take a look at this Smiley Button project, or this more advanced Voting Machine project, both controlled using the buttons in JavaScript.

Pins



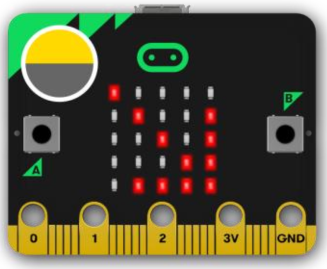
What is it? There are 25 external connectors on the edge connector of the micro:bit, which we refer to as 'pins'. Program motors, LEDs, or other electrical components with the pins, or connect extra sensors to control your code!

How do I code it? Learn more about the hardware of the pins, or use the code references below.

Python or MakeCode

Examples - code a Banana Keyboard, hack your headphones, and create a Milk-Carton Robot with JavaScript! Or, take a look at this ticklish micro:bit project in Python!

Light Sensor



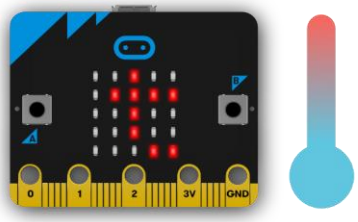
What is it? By reversing the LEDs of the screen to become an input, the LED screen works as a basic light sensor, allowing you to detect ambient light.

How do I code it? See the code references below.

Python or MakeCode

Examples - learn how to chart the light level on the screen with JavaScript

Temperature Sensor



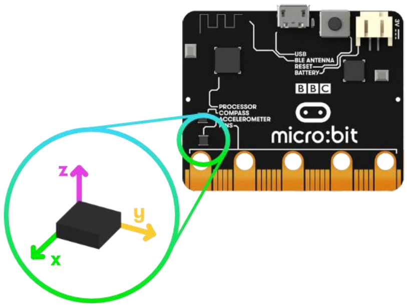
What is it? This sensor allows the micro:bit to detect the current temperature of the device, in degrees and Celsius.

How do I code it? See the code references below.

Python or MakeCode

Examples - discover how the temperature sensor works.

Accelerometer



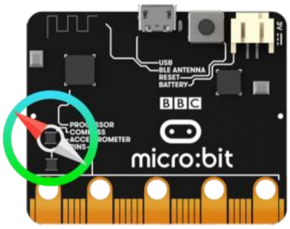
What is it? An accelerometer measures the acceleration of your micro:bit; this component senses when the micro:bit is moved. It can also detect other actions, e.g. shake, tilt, and free-fall.

How do I code it? See the code references below.

Python or MakeCode

Examples - code a Rock, Paper, Scissors game with JavaScript, triggered when the micro:bit is shaken! Or, create musical mayhem with Python!

Compass



What is it? The compass detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used.

'Calibrating' the compass ensures the compass results are accurate. For the MakeCode editor, use the 'calibrate compass' block. To calibrate the compass in Python use compass.calibrate().

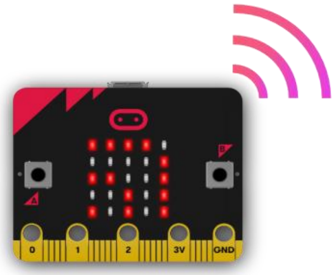
When the calibration begins, the micro:bit will scroll the instruction "Tilt to fill screen". To calibrate the compass, tilt the micro:bit to move the dot in the centre of the screen around until you have filled up the whole screen.

How do I code it? Use the code references below.

Python or MakeCode

Examples - create a working compass to find North in JavaScriptor Python!

Radio



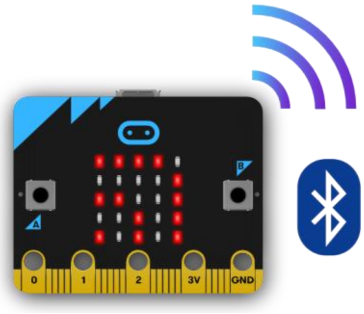
What is it? The radio feature allows you to communicate wirelessly between micro:bits. Use the radio to send messages to other micro:bits, build multiplayer games, and much more!

How do I code it? Discover how to code the radio:

Python MakeCode

Examples - create a Multiplayer Rock, Paper, Scissors game (JavaScript), or create cool digital fireflies in JavaScript and Python!

Bluetooth



What is it? BLE (Bluetooth Low Energy) allows the micro:bit to control phones and tablets over Bluetooth. This communication works both ways, so you can also send code to your micro:bit wirelessly from your phone using one of our apps. Other apps, such as Swift Playgrounds and Scratch, use Bluetooth to talk to the micro:bit.

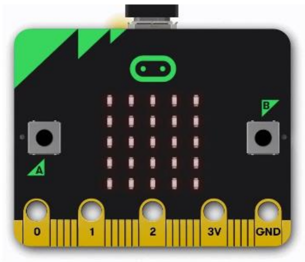
Before using the Bluetooth feature you will need to pair your micro:bit with another device. Once paired, you can send programs wirelessly to your micro:bit. When you're using Radio, Bluetooth can still be used to update the code on your micro:bit, if you enter pairing mode - learn about the differences between radio and bluetooth in this support article.

MakeCode

The Python Editor doesn't currently support bluetooth.

What can I do with it? Send code to your micro:bit wirelessly.

USB Interface



What is it? The USB interface allows you to connect the micro:bit to your computer via a micro-USB cable, which will power the device and allow you to download programs onto the micro:bit.

Raspberry Pi

What is the Raspberry Pi?

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

How does the Raspberry Pi work?

An SD card inserted into the slot on the board acts as the hard drive for the Raspberry Pi. It is powered by USB and the video output can be hooked up to a traditional RCA TV set, a more modern monitor, or even a TV using the HDMI port.

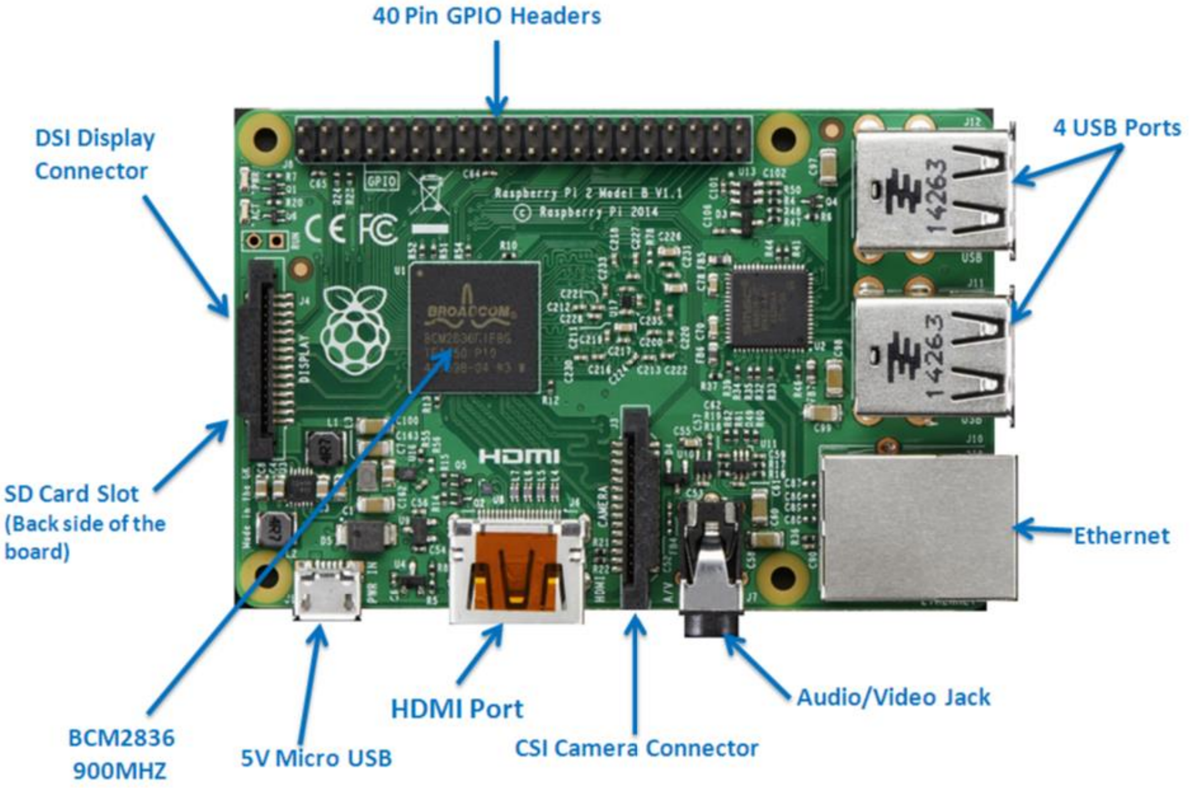
What's the point of a Raspberry Pi?

The Raspberry Pi is a micro-computer initially designed for education. It has all of the components you would see on a normal family desktop PC — a processor, RAM, HDMI port, audio output and USB ports for adding peripherals like a keyboard and mouse.

What is the advantage of Raspberry Pi?

Although the Raspberry Pi is as small as the size of a credit card, it works as a normal computer at a relatively low price. it is possible to work as a low-cost server to handle light internal or web traffic. Grouping a set of Raspberry Pi to work as a server is more cost-effective than a normal server.

Raspberry Pi components



- **ARM CPU/GPU** -- This is a Broadcom BCM2835 System on a Chip (SoC) that is made up of an ARM central processing unit (CPU) and a Videocore 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- **GPIO** -- These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- **RCA** -- An RCA jack allows connection of analog TVs and other similar output devices.
- **Audio out** -- This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
- **LEDs** -- Light-emitting diodes, for all of your indicator light needs.
- **USB** -- This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.
- **HDMI** -- This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- **Power** -- This is a 5v Micro USB power connector into which you can plug your compatible power supply.
- **SD card slot** -- This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from

the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.

- **Ethernet** -- This connector allows for wired network access and is only available on the Model B.

Many of the features that are missing, such as WiFi and audio in, can be added using the USB port(s) or a USB hub as needed. Furthermore, the picture above may not show all the features since there are many different Raspberry Pi's.

What can I do with a Raspberry Pi?

- Desktop PC
- Wireless Print Server
- Media Center
- Retro Gaming Machine
- Minecraft Game Server
- Robot Controller
- Stop Motion Camera

Module: Microcontrollers

Topic: Logic Gates

General Description:

Most electronic systems have one or more inputs and one output that can only work correctly if it follows a logical pattern. In order to manipulate these inputs and outputs represented by 1's and 0's, electronic systems use logic gates to perform basic logical functions. Therefore, logic gates can be seen as the fundamental blocks to make a digital integrated circuit work. In other words, without logic gates electronic equipment (such as integrated circuits, CPUs for computers, smartphones, and smart TVs) would not work.

While there are basic circuits that only use a few logic gate functions, others, such as microprocessors, use millions of them.

This handout provides guidelines that will allow you to understand the seven different types of logic gates, as well as how to operate and create circuits with logic gates through practical and simulated assembly.

Learning Objective:

Handout learning objective:

- Understand how logic gates work and be able to identify them (AND, OR, XOR, NOT, NAND, NOR, XNOR);
- Use a simulator app to test a logic design and debug it;
- Recognize the importance and applicability of using logic gates;
- Understand how digital circuits come together to form complex, computer systems.

Digital skills learning objective:

- 1.1, 1.2
- 3.1, 3.2, 3.4
- 5.1, 5.2, 5.3, 5.4

Preparatory steps for the trainer:

- Prepare a brief introduction on logic gates to help participants understand its importance in modern technology;
- Familiarise yourself with the 7 logic gates, how they work and how to plot truth tables given a logic gate diagram;
- Select a smartphone game that participants can play and debug and test a logic gate design (e.g. Circuit Scramble APK; Logic Gate Simulator; Smart Logic Simulator; Logic Circuit Simulator Pro)
- Familiarise yourself with the following chips: NE555; 74192 and 7448.

Exercises:

- Task sheet A2.1 – Logic Gates: AND, OR
- Task sheet A2.2 – Logic Gates: AND, NOT
- Task sheet A2.3 – 7-segment Display 0 to 9 counter

How to organise your learning:

- Explain the 7 logic gates symbols and give examples on how to use them; A simple video on Logic Gates can also be shown. E.g.: AND, OR, NOT gates Video and NAND, NOR, XOR, XNOR gate Video; Furthermore, explain how to plot truth tables following a given diagram: Logic Gates and truth tables.
- It could be useful to point out the relevance between some gates in order to help memorise their tables and avoid confusion. For example, NAND and AND gates have supplemental outputs for identical inputs. Same applies for OR and NOR.
- For some technical background information, explain the concept behind logic gates by using specific real-life applications. For instance:
  - A simple light switch using the logic gate OR: in one position of the switch, the value is OFF and equals 0, while in the other position the value is ON and equals 1. Imagine a room with two light switches (A and B) that can be used to light up the same lamp. If both switch A and B are off, the output is 0 and so the light will be turned off. However, if either switch A OR switch B are ON, then the final output is 1 and the light will be lit.

    The following truth table sums it up:

Switch A	Switch B	Is the lamp On or OFF?
OFF (0)	OFF (0)	OFF (0)
OFF (0)	ON (1)	ON (1)
ON (1)	OFF (0)	ON (1)
ON (1)	ON (1)	ON (1)

  - An anti-theft system using the logic gate AND: Imagine that this device is formed by 1) the alarm system, and 2) a movement sensor. Let us consider that this movement sensor is always ON (1). Now, if the alarm system is OFF (0) and a burglar shows up, the anti-theft system will not start because the final output is 0. For the anti-theft system to initiate (meaning the output is 1), both the alarm system AND the movement sensor inputs' must be 1. That is, if the alarm system is ON (1) AND the movement sensor is also

ON (1), when a burglar shows up, the alarm will ring and the police will be notified.

The following truth table sums it up:

Alarm system	Movement sensor	Will the alarm ring?
ON (1)	ON (1)	YES (1)
OFF (0)	ON (1)	NO (0)
ON (1)	OFF (0)	NO (0)
OFF (0)	OFF (0)	NO (0)

- For Task sheet A2.1 (Logic Gates: AND, OR) and Task sheet A2.2 (Logic Gates: AND, NOT), have participants make the logic gate functions exercises and then play “Circuit Scramble – Computer logic puzzles”, or any other Logic Gate game, during 20 to 30 minutes, so that they can test what they have learned.
- Assembly exercises should be done after Task sheet A2.1 (Logic Gates: AND, OR) and Task sheet A2.2 (Logic Gates: AND, NOT). Afterwards, assembly exercise of Task sheet A2.3 (7-segment Display Counter) can also be done. Preferably, in order to avoid unnecessarily damaging any electronic component, the trainer can demonstrate participants how to build the activity on Task Sheet A2.3 and participants can try to replicate it.

Additional information:

- Learn more about logic gates: https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR
- Creation of diagrams and the correspondent truth table: https://www.youtube.com/watch?v=BnB2m1nXZ84
- Making a logic gate from transistors: https://www.youtube.com/watch?v=sTu3LwpF6XI

Module: Microcontroller

Topic: Introduction into (Block)Coding

General Description:

Block Coding is a good way for beginners to enter into the coding world. The code exists as blocks and can be implemented by dragging a code block into the programme.
This helps the participants to focus more on the logic and the basics of programming instead of getting overwhelmed by the correct usage of syntax and functions. The task sheets were implemented using a micro:bit and Calliope mini but any microcontroller can be used as long as it has buttons and LEDs.

Learning Objective:

Handout learning objective:

- Understand the basics of programming with the help of block coding
- Understand the basic logic of programming
- Understand the transfer between block coding and script coding
- Be able to implement simple commands
- Learn about various application possibilities
- Be able to implement little projects with micro:bit

Digital skills learning objective:

- 1.1, 1.3
- 3.1, 3.2, 3.4
- 5.1, 5.2, 5.4

Preparatory steps for the trainer:

- The trainer familiarises her- or himself with the use and application possibilities of the micro:bit microcontroller https://microbit.org/guide/quick/ or whatever microcontroller the trainer is using
- A block building software for several microcontrollers you can use: https://ide.mblock.cc/#/. You need to prepare the circuits with the Arduino and then program the Arduino.

Exercises:

Note: These task sheets are made for microcontrollers that have buttons and LEDs on them. Some of these task sheets may not work if you are not using such a microcontroller. Furthermore, the coding blocks may differ a bit between different microcontrollers.

- Task sheet A3.1 – Programming a click counter
- Task sheet A3.2 – Programming a mini piano
- Task sheet A3.3 – Tilt the microcontroller to light the LED up
- Task sheet A3.4 – Programming the game “Rock, paper, scissors”
- Task sheet A3.5 – Mental arithmetic vs. a microcontroller
- Task sheet A3.6 – Programming the game “Hot Potato”
- Task sheet A3.7 – Programming the game “Avoid the obstacle”
- Task sheet A3.8 – Programming a game “Catch the LED”
- Task sheet A3.9 – Block coding practice with a pre-made circuit

If you worked with an Arduino you can use these challenges:

E-DESIGN - Blockcoding Challenges_ENG

How to organise your learning:

- Continue with a little theory input and give the participants the theory they need to solve the task sheets for that lesson.
- The task sheets are in order. The first task sheets come with a lot of help. As you get further with the task sheets this help is reduced more and more. Finally, you will reach the task sheets that are without visual help and text only so that the participants have to develop their own logic and structure.
- You can use the following examples to teach some technical background (not obligatory):
  - The On/Off button represents the 1 and 0 from the binary system. Most people do not know why it looks the way it looks. Use it to introduce the binary system to the participants.
  - A remote sends infrared signals. It is not visible to the human eye but you can see the laser with a camera (e.g., from a smartphone). Use it to introduce wavelengths and frequencies so that the participants can understand how a speaker and its frequencies work.

- Every smartphone has sensors in it. Most participants do not know what kind of sensors are integrated in their smartphones. It is possible to have a look at them. Many smartphones (but not all) support this. Just enter the Code ***#0*#** into your dial app. The menu for all sensors and functionalities will open up. The sensors are the same sensors the participants can use with Arduino or other microcontrollers.

Additional information:

- More information about microcontrollers can be found here:

  Module: Microcontroller - Topic: Introduction into microcontrollers

Module: Microcontrollers

Topic: Arduino Basics

General Description:

Arduino is a platform that allows sending and receiving information from practically any other electronic device. Thus, it enables, for instance, the creation of a data collection system of sensors (such as temperature, illumination control, distance, etc.) which will afterwards be processed and can be sent to a remote system or display them on a screen.
Following this premise, this handout will provide guidelines to understand how sensors can be connected to the Arduino and how to test them through simulation software and hands-on work.

Learning Objective:

Handout learning objective:

- Understand how microcontroller Arduino boards work;
- Use, test and identify different types of sensors;
- Build a simple Arduino project using sensors;
- Perform sensor and motor assemblies.

Digital skills learning objective:

- 1.1, 1.3
- 2.4
- 3.4
- 5.1, 5.2, 5.3, 5.4

Preparatory steps for the trainer:

- Familiarise with the following sensors: PIR sensor (task sheet A4.2), ultrasonic sensor SRF 05 ultra and sharp distance sensor (task sheet A4.3).

Exercises:

- Task sheet A4.0 – Learning if/else-query and while-loop
- Task sheet A4.1 – Controlling a LED using a push button
- Task sheet A4.2 – Connecting a PIR sensor to an Arduino
- Task sheet A4.3 – Connecting an ultrasonic sensor to Arduino
- Task sheet A4.4 – Creating an air conditioning system using block code

How to organise your learning:

- The simulation software used, Tinkercad, requires a Google Account registration, therefore the trainer must make sure everyone has access to it. Once logged in, the trainer should explain how the platform works: how to perform the assembly of their circuits, search for components, how to connect them, etc.; Furthermore, the trainer should explain the concept behind the Arduino microcontroller, i.e. how it works and its potential in the electronics world;

- Task sheet A4.1 (Controlling a LED using a push button) is the most basic task sheet. The trainer should encourage participants to solve the task on their own, relying to online sources if needed. As for Task sheet A4.2 (Connecting a PIR sensor to an Arduino) and Task sheet A4.3 (Connecting an ultrasonic sensor to Arduino), they are intermediate, so the trainer's support will be needed, especially if participants do a physical assembly.
- Since it is possible to model objects on Tinkercad, this handout can be combined with 3D Modelling and 3D Printing task sheets. For instance, participants can model a small box with holes and print it, so that they can afterwards place the result of each Microcontroller task sheet inside the box and make it look cleaner (all cables are hidden, only a push button is visible, for instance etc.);
- Although all task sheets can be completed using a simulation software, it is encouraged that participants do the physical assembly as well, so as to put theory and simulation in practice; During physical assembly, the trainer should assist participants in connecting and programming sensors so as to avoid any unnecessary damage.

Additional information:

- **Arduino in 15 minutes:** https://www.youtube.com/watch?v=nL34zDTPkcs
- **15 Great Arduino Projects for beginners:** https://www.youtube.com/watch?v=Ox-9eOc3bQU
- **Arduino Workshop for beginners with videos:** https://core-electronics.com.au/tutorials/arduino-workshop-for-beginners.html

Module: Microcontroller

Topic: Introduction into IoT

General Description:

The Internet of Things (IoT) is a network of 'smart' devices that connect and communicate via the Internet. The key to the IoT is the interconnectivity of devices, which collect and exchange information through embedded software, cameras and sensors which sense things like light, sound, distance and movement without requiring human-to-human or human-to-computer interaction.

An IoT system consists of sensors/devices which “talk” to the cloud through some kind of connectivity. Once the data gets to the cloud, software processes it and then might decide to perform an action, such as sending an alert or automatically adjusting the sensors/devices without the need for the user.

The handout is intended to give a rough overview of the IoT, its uses and to give the possibility to the students to build IoT projects.

Learning objective:

Handout learning objective:

- Understand the meaning, the potential and possible applications of the IoT
- Be able to create a simple IoT project (like using sensors to record temperature and push notifications)

Digital skills learning objective:

- 1.1
- 2.1, 2.2, 2.3, 2.5
- 3.1, 3.2, 3.4
- 4.1, 4.4
- 5.1, 5.2, 5.3, 5.4

Preparatory steps for the trainer:

- Familiarise yourself with the programme and hardware you want to use

  (Software: Arduino IDE, Hardware: ESP32/ESP8266, ESP32, DHT11, BMP180, Raspberry Pi)

Exercises:

Suggested Projects for IoT:

- Task sheet A5.1 – Arduino weather station
- Task sheet A5.2 – Create your local server
- Task sheet A5.3 – Sensing the environment & Notifying

How to organise your learning:

For helping the participants to understand IoT and how it can be used, you could offer them a few real-life examples where IoT is used. Such examples could be:

Wearable health monitors: Smartwatches and other wearable devices for monitoring health status (pressure, blood oxygen, heart rate etc.) are becoming more popular each day. The devices can report an emergency like an asthma attack, heart failure, etc., immediately to a physician. Data is stored online and can be accessed anytime by a physician. IoT automates the workflow by allowing the provision of effective health care services to the patients. See here for some examples.

Smart IoT farming applications: The devices can detect weather conditions and other environmental data and through the applications can help farmers to optimise a lot of different activities such as determining the best time to harvest plants, creating fertilizer profiles based on the chemistry of soil, and sensing soil nutrients and moisture levels. See here for some examples.

Smart home security systems: Home security systems with integrated cameras, smart locking mechanisms, and remote controls allow homeowners to monitor what is going on inside and outside, or to see and talk to visitors from miles away. See here for some examples.

In order to improve the quality of the course and its outcomes for the participants this handout could be used after the Web Development handout, so the participants are already familiar with the idea of the internet and how data is stored or served. The participants familiarise themselves with basic aspects of web development (it is especially useful for the exercise Sensing the Environment _ Notifying) and then they can proceed to the more complicated aspects of IoT and microcontrollers, such as Blockcoding or Logic gates for better understanding of programming and IoT.

Additional information:

Other possible exercises can be found here:

- Beginner:
  - https://www.hackster.io/raspberry-pi/projects?difficulty=beginner
- Intermediate
  - https://www.hackster.io/raspberry-pi/projects?difficulty=intermediate
- Expert
  - https://www.hackster.io/raspberry-pi/projects?difficulty=hardcore

More Useful links:

- Free education material and courses:
  - https://www.coursera.org/browse/information-technology/cloud-computing (Free only for auditing the courses)
  - https://alison.com/course/internet-of-things-and-the-cloud
- You can also check out videos on YouTube explaining IoT and showcasing its uses

  - IoT Full Course: https://www.youtube.com/watch?v=h0gWfVCSGQQ
  - Internet of Things Architecture for Beginners: https://www.youtube.com/watch?v=KeaeuUcw02Q

Module: 3D Objects

Topic: 3D Modelling

General Description:

3D modelling is the process of generating a 3D image of any object using a Computer-Aided Design (CAD) software. Afterwards, the object can be 3D Printed or CNC machined.
This handout will provide guidelines to design objects using 3D modelling software. In order to achieve this, participants will learn how to use CAD software, as well as other important aspects related to 3D modelling, such as designing simple objects based on real objects, learning to use different techniques to mate components and learning how to assemble them.

Learning Objectives:

Handout learning objectives:

- Create a 3D object using 3D CAD software;
- Be able to identify measuring tools and know how to use them;
- Design an object using 3D software, based on the proportions of other objects (that is, a virtual replication/copy);
- Assemble different components on CAD software;
- Be able to identify the potential and possible application fields of CAD software in the real-world context.

Digital skills learning objectives:

- 3.1, 3.2
- 4.4
- 5.1, 5.3, 5.4

Preparatory steps for the trainer:

- There are various free CAD software available that can be used to teach 3D modelling (e.g. Tinkercad, Sketchup Online, and Onshape). To start off, the trainer should select which software will be used and be sure to be familiar with it. Some introduction videos follow:
  - Tinkercad (beginner): https://www.youtube.com/watch?v=4KpvAcZ2wAo
  - SketchUp (intermediate): https://www.youtube.com/watch?v=0cmI-XLgyoM
  - OnShape (advanced): https://www.youtube.com/watch?v=pMWnsHpDlQE

Exercises:

- Task sheet B1.1 – Model and edit 3 simple objects – Part 1
- Task sheet B1.2 – Model and edit 3 simple objects – Part 2
- Task sheet B1.3 – Assembly of a bench vise
- Task sheet B1.4 – Assembly of a wheel

How to organise your learning:

- Introduce the topic of 3D modelling to participants by raising awareness on the importance of CAD in the real world. To do so, mention its use in different areas (eg. in the automotive and aircraft industry, architecture, interior design, product design, computer games industry, film industry, virtual reality etc.). For a more meaningful understanding show different videos in which 3D CAD is used:
  - Aerospace industry: https://www.youtube.com/watch?v=q9YQ0jxCXGc
  - Architecture: https://www.youtube.com/watch?v=MdPPeMSTO30
  - Virtual reality: https://www.youtube.com/watch?v=mWaQfjEJIMQ
- Present the software that you have selected and have participants explore the software (Tinkercad is a good suggestion to start with since it is very user-friendly). After demonstrating how to use some basic commands, have participants model simple geometric shapes as a tryout.
- Task sheet B1.1 (Model and edit 3 simple objects – Part 1) or Task sheet B1.2 (Model and edit 3 simple objects – Part 2) should be part of their first exercise once they have explored the software. In these two task sheets, participants need to model objects according to given dimensions, therefore it will take some time. Task sheet 2 is slightly more challenging than task sheet B1.1, therefore, if participants are already familiar with 3D modelling, you can stick to task sheet B1.2 only. Otherwise, make sure they model all 6 objects from both Part 1 and Part 2;
- Similarly to task sheets B1.1 and B1.2, task sheets B1.3 (Assembly of a bench vise) and B1.4 (Assembly of a wheel) are assembly exercises that can be implemented alternatively.
- To make the 3D Modelling experience more relevant, if possible, have participants print examples of objects that they have designed. This way, if participants want to further develop the topic of 3D Modelling, the 3D Printing Handout should come next.

Additional information:

- What is 3D Modelling & how is 3D modelling used: https://homesthetics.net/what-is-3d-modeling/
- You can draw ideas for other projects, from 3D modelling to assembling, at: www.thingiverse.com
- Other useful free 3D modelling software:

  http://www.meshmixer.com/

  https://www.meshlab.net/

  https://all3dp.com/1/free-3d-modeling-software-free-3d-design-software-3d-cad/

Module: 3D Objects

Topic: 3D Printing

General Description:

This handout will give an overview of how to work with the different functionalities of a 3D printer and how to use it to print different objects. The term "3D printing" can refer to a variety of processes in which material is joined or solidified under computer control to create a three-dimensional object with the material being added together, typically layer by layer.

One of the key advantages of 3D printing is the ability to produce very complex shapes or geometries that would be otherwise impossible to construct by hand, including hollow parts or parts with internal truss structures to reduce weight.

Learning objectives:

Handout learning objective:

- Understand how a 3D printer works
- Have a basic understanding of slicer software (e.g. “Ultimaker Cura”)
- Be able to print 3D objects independently and scaling the object with given measurements/settings
- Know how to upload STL files of 3D objects on the SD card

Digital learning objective:

- 1.1
- 2.1, 2.2
- 5.1, 5.2, 5.3, 5.4

Preparatory steps for the trainer:

- As for slicing software, we recommend Ultimaker Cura
- Select a suitable 3D object for a test print (paying attention to a relatively short printing time of the object)
- Further trainer information:
  - Video tutorial: Ultimaker Cura 3D Slicer For Beginners

Exercises:

- Task sheet B2.1 – Testing and calibration
- Task sheet B2.2 – Changing the filament
- Task sheet B2.3 – Preheating and Cooling
- Task sheet B2.4 – Print an object

How to organise your learning:

The trainer should:

- explain the basic operation of the 3D printer and the functions of the slicer software
- give instructions on the pressure settings selected by the participants so that participants reproduce what they have learned
- have groups/participants analyse the printing settings of an already printed object and check whether the settings gave the desired result
- explain to participants how to manage/calculate the time different objects with different dimensions take to print
- follow from the 3D modelling handout, and can lead to assembly.

3D-Printing in Real Life

- 3D printing is used in a number of real-life settings. Since this technology has experienced increased innovation and advancement in the last years, it is now practically applied in a number of settings:
    - 3D Printing is used for prosthetic body parts in the context where animals or humans have suffered an injury (Common 3D-printed prosthetics)
    - Used in home and building construction as it is very quick - i.e. can be very useful in case of a natural disaster in terms of creating emergency shelters in much less time (3D Printing - The Future of Construction)
    - Used for printing common everyday objects - decorations, musical instruments, edible food (chocolate, ice cream), household items, etc. (3D-Printed Instruments)
    - 3D printing is currently largely used in the manufacturing sector - printing bodies of cars and electric vehicles, replacement parts of machinery, prototyping new products in almost all labour market sectors and professional fields, and creating jigs and moulds to improve the production process efficiency. (3D Printing for Aerospace & Defense)

Additional information:

- Other possible exercises:
    - Print a 3D object with multiple colours
    - Print pen cases
    - Print functional wrenches
    - Print a mini monument
    - Print a ship

- 3D object platforms:
  - www.thingiverse.com
  - https://all3dp.com/1/free-3d-models-download-best-sites-3d-archive-3d/ (here you can find further 3D object platforms)

- For self-study:
  - 3D extensive and multilingual *online learning platform* on the subject of 3D printing: http://e3dplus.cetemlearning.eu/
  - Video tutorial on how to use Ultimaker Cura 3D Slicer For Beginners

Module: 3D Objects

Topic: Photogrammetry

General Description:

Photogrammetry is a method of indirectly determining an object's position and shape by using 2D photographs. In photogrammetry, the focus is on the exact three-dimensional geometric reconstruction of the photographed object. By interpolating many 2D pictures from different angles and perspectives, a digital model of the photographed object (or a landscape) is obtained (if the photogrammetry method is used correctly).

The principles of photogrammetry can be applied to various fields, technologies and sciences: e.g. architecture, art, geology, agriculture, forestry and biomedical applications. Even applications such as Google Earth would be unthinkable without the photogrammetry method.

A practice-oriented introduction to this topic is intended to give the course participants an insight into the potential and the basic functioning of the procedure and enable them to gain experience with this exciting topic.

Learning Objective:

Handout learning objective:

- Participants understand the basic principles of photogrammetry
- Participants have an idea of different application possibilities of photogrammetry
- Participants can create and edit (modify, combine, repair, simplify, change size and orientation) 3D objects by processing photos through photogrammetry software

Digital learning objective:

- 1.1,
- 2.2, 2.4 *(in case the class uses cloud storage services to share their photogrammetry results)*
- 3.1, 3.2, 3.3,
- 5.1, 5.3

Preparatory steps for the trainer:

Necessary software[1]:

- **3D Flow Zephyr (recommended)**
  - Download: https://www.3dflow.net/3df-zephyr-free/
  - Very simple to use and good documentation and tutorials: https://www.3dflow.net/technology/documents/3df-zephyr-tutorials/
  - Disadvantage: in the free version you can process only 50 images

or

[1] Here you find a list of: "Best Photogrammetry Software 2020": https://all3dp.com/1/best-photogrammetry-software/

- **Visual FSM**
  - Download: http://ccwu.me/vsfm/
  - Important: “CMVS-PMVS” is needed too.
    Download: https://github.com/pmoulon/CMVS-PMVS
  - Trainer suggestion: Provide participants with the software ready prepared for use as a download link - for example via Gdrive.
  - Tutorial how to install and use: https://www.youtube.com/watch?v=GEAbXYDzUjU

- The trainer familiarizes her- or himself with the use and application possibilities of the used software.
- The trainer should test the software beforehand.
- It is recommended to show the participants both suggested programs. The advantage of 3D F Zephyr is the user-friendly handling and the good software documentation. Visual FSM, on the other hand, makes it possible to watch in real time when processing the point clouds, which is impressive to see.
- The processing of 3D objects takes a long time and depends on the performance of the computer used and also on the number of images. 30 to 50 images should give good results for illustration. In general, the better the quality of the photos and the higher the number of usable photos of the object, the better the results will be.
- 2 further useful basic tutorials to improve your results:
  - https://www.youtube.com/watch?v=LeU_2SHwhqI
  - https://www.youtube.com/watch?v=diQAJO4sghQ
  - https://www.youtube.com/watch?v=9_F-b2hxP_o

Exercises:

Basic:

- Task sheet B3.1 – Getting Started (Introduction to Photogrammetry)
- Task sheet B3.2 – 3D scan an object
- Task sheet B3.3 – 3D scan editing / mesh creation

Pursuing:

- Task sheet B3.4 – Using the masking tool (to improve the scan results) (3D Flow Zephyr)

How to organize your learning:

- The trainer introduces participants to Photogrammetry. It is recommended to outline the historical development as well as today's purposes of use:
  - The trainer can start by pointing out that the origins of photogrammetry were in building surveying and especially in cartography:
    - The "Stereoautograph" invented by Eduard von Orel in 1907, e.g. made it possible for the first time to draw the height layer lines of a map automatically by optical sensing of stereo image photographs.

        - In the 1930s, Paul Gast developed the "Bundle Adjustment", which has made it possible to use this method with computer support since the 1960s.
        - For this purpose, a 2-minute video is recommended that illustrates the historical/technical development of P. well: Evolution of Analog to Digital Mapping
    - In the next step, the participants could discuss together where they see examples of up-to-date applications.
    - The trainer can then present further examples from the fields of architecture, art, geology, agriculture, archaeology, game design, cartography, applications like “Google Earth” and many more.
- The trainer demonstrates how to use the chosen software
- Participants watch again a tutorial about how to use the software and make some notes (if necessary)
- Participants do exercises on objects they like to scan
- The trainer can decide on the working group size of participants (in accordance with available equipment)
- Participants can also practice at home (with a mobile phone camera or other cameras)
- The Photogrammetry activities can be perfectly combined with 3D printing activities: the scanned objects can be post-processed in order to create a 3D printable mesh file

Additional information:

- Article: https://en.wikipedia.org/wiki/Photogrammetry
- Article: http://culturalheritageimaging.org/Technologies/Photogrammetry/
- Video: Definition and History of Photogrammetry https://www.youtube.com/watch?v=ixQSQnFryM8
- Video: How to take the right pictures - Image collection basics
- **7 Photogrammetry Rules:**
    - Set the manual exposure mode
    - Use a tripod; Avoid shaky hands
    - Get enough light (open curtains etc.)
    - Ambient light is good; Hard shadows may cause problems
    - No transparent material and super reflective surface
    - Use a long-duration shutter speed if not enough light
    - Photograph the object from all possible angles
- Further tips:
    - Use a good camera with a high resolution and a large sensor
    - Use an uncompressed recording mode to get better results
    - Processing lots of pictures requires a long processing time if there is not enough CPU/GPU capacity

Module: App Development

Topic: Introduction to MIT App Inventor 2

General Description:

Different platforms can be used to teach the concepts of programming through the development of mobile applications. This handout will focus on MIT App Inventor 2 and will provide guidelines to develop basic software apps for Android.

MIT App Inventor 2 is appropriate for individuals who are new to computer programming because it allows to use built-in block codes (a piece of code already grouped together) to create an algorithm and trigger different events. To do so, one just needs to drag and drop components to create the visuals, and then the built-in blocks to program the application behaviour of the components selected.

MIT App Inventor 2 requires a Google Account registration; therefore, users must have a Gmail account created.

Learning Objective:

Handout learning objective:

- Understand the concept and logic behind computer programming;
- Identify and explain the concept of coding-blocks;
- Acquire basic programming knowledge on app development;
- Compile and test a simple application using MIT App Inventor 2;
- Be able to recognise the advantages and the role played by apps and understand how can they be applied to different real-world contexts and identify the need and opportunity in app markets;

Digital skills learning objective:

- 2.1, 2.2
- 3.1, 3.2, 3.4
- 5.1, 5.2, 5.3, 5.4

Preparatory steps for the trainer:

- Familiarise yourself with App Inventor 2 and a smartphone emulator (e.g. Bluestacks)
  - **MIT App Inventor 2:**
    - Beginner tutorial set of videos
    - Intermediate tutorial set of videos
  - **Bluestacks:**
    - MIT App Inventor Simulator
    - Using BlueStacks Android Emulator with MIT APP Inventor
    - Android Studio Emulator

Being familiar with an android emulator can be particularly useful if any of the participants do not have a smartphone. This way, created apps can still be tested using a computer;

- MIT App Inventor 2 requires a Gmail account to login, so make sure participants have one.

Exercises:

- Task sheet C1.1 – Show and hide an image
- Task sheet C1.2 – Talk to me
- Task sheet C1.3 – Random colour generator

How to organise your learning:

- Make a brief introduction about smartphone apps to help participants understand its importance in the current job market.
- Show participants different apps with different functionalities, starting with basic ones. (you can check the beginner and intermediate tutorial set of videos above, for different ideas). While displaying, introduce them to different and new functions.
  Some examples of simple apps that can be shown are: Talk to me app; Random colour generator; For loop; Input value; Procedure;
  Intermediate app examples: Button and components visibility, Layout Arrangement, Counting a number;
  Showing different apps is important because participants can feel inspired, and might want to try to replicate them or come up with their own ideas for apps.
- To better explain how MIT Inventor 2 works, the trainer can select one of the apps and show participants how to create it from scratch. Afterwards, ask participants to do task sheet C1.1 (Show and hide an image) and then test the app using their smartphone or a computer (with a smartphone emulator); If they succeed, they can move to task sheet C1.2 (Talk to me) and then task sheet C1.3 (Random Colour Generator).
- When participants feel at ease, challenge them to build an app using some of the functions that they learnt, or possibly improve the ones that they have created in the task sheets by exploring online sources and using new functions.

Additional information:

- Not sure what kind of app to create? You can draw some ideas and inspirations from https://appinventor.mit.edu/explore/ai2/tutorials.html

- Have some doubts about MIT App Inventor 2? Check the Frequently Asked Questions: https://appinventor.mit.edu/explore/content/faq

Module: Web development

Topic: HTML/CSS/JavaScript

General Description:

Web development is based basically on HTML, CSS and JavaScript programming. HTML language is used to create the content and the structure of the webpage, CSS is the language that defines the visual part and JavaScript is a script language that allows the developer to change the content dynamically and give interactivity to the website.

This handout includes information on the basic theory behind web development (i.e. how the internet works) and gives many possibilities to let students explore HTML, CSS and JavaScript in practice. The task sheets include activities for beginners and for more advanced learners, and include the creation of a web page based on a visual example, task based on understanding the code of a more complex website, and different coding challenges to teach more advanced code writing such as <div> and meta codes to structure websites.

Learning Objective:

Handout learning objective:

- Understand how the connection between server and client on the internet works
- Read, understand and modify an HTML & CSS code (basic web languages)
- Increase the knowledge of the different tools, methods, and frameworks to create a website
- Able to apply the basics of web development language and code to use more complex tools for the development of websites.
- Use the different HTML tags and CSS rules in the creation of simple web page
- Learn how to use some basic JavaScript in order to add dynamic content and create interactive web pages.
- Being aware of the increasing importance and uses of web-development on the labour market

Digital skills learning objective:

- 1.1, 1.2, 1.3
- 2.1, 2.2, 2.3, 2.4
- 3.1, 3.2, 3.3, 3.4
- 5.1, 5.2, 5.3, 5.4

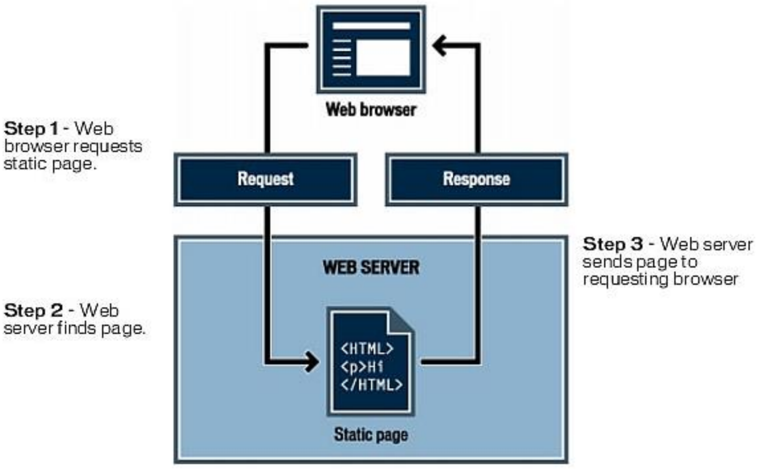
Preparatory steps for the trainer:

- Theoretical background:

These are the most important general concepts and ideas to introduce to the students

Client-Server relation

Understanding how webs are serviced through the internet is important in order to understand how to create a webpage or service on a server. Based on a Client - Server structure, all web pages are resources that are stored on an online server. When the user accesses a URL or internet address, it will be asking the server a certain resource, for example, a web page, that will be offered and shown on a browser.


Web browser
Step 1 - Web browser requests static page.
Request
Response
WEB SERVER
Step 3 - Web server sends page to requesting browser
Step 2 - Web server finds page.
<HTML>
<p>Hi
</HTML>
Static page

For more information about the client-server connection, please look here: https://www.computerhope.com/jargon/w/webpage.htm

- The trainer should also familiarize with the different programming languages normally used for web development. These are some useful links where you can get more information about these topics:
    - More information + improve coding skills in HTML, CSS and JavaScript with online courses or editors:
        - https://www.w3schools.com/
        - https://www.codecademy.com/learn/learn-html
        - https://www.codecademy.com/learn/learn-css
        - https://www.codecademy.com/learn/introduction-to-javascript
        - https://codedoor.com/
        - https://www.khanacademy.org/computing/computer-programming/html-css

Exercises:

The following task sheets can be used for all learner levels. When students are more advanced, add more advanced coding challenges such as meta structures including CSS and HTML or JavaScript.

- HTML and CSS code:
    - Task sheet D1.1 – Create your first web page
    - Task sheet D1.2 – Create your first webpage using a visual block editor
    - Task sheet D1.3 – Find the errors
    - Task sheet D1.4 – Creating a website based on a specific target group (including design thinking)
- JavaScript:
    - Task sheet D1.5 – Make your web page interactive with javascript
    - Task sheet D1.6 – Create your first interactive e-commerce
- PHP:
    - Task sheet D1.7 – Create a working contact form

How to organize your learning:

Start explaining to the participants the many uses of web development nowadays. Show the participants some general examples online of different useful website types as personal pages, blogs, e-commerces, social networks, etc. After this, make emphasis that web development, thanks to the evolution of the network speed and technologies is also used in internal web applications as companies' intranets, file hosting services, and even small web applications for smart homes, like security control systems or home automation systems linked to their personal mobile phones to receive notifications.

It is important to focus on these examples in order to widen the vision of the participants in the different uses that web development can have.

Continue with a brief explanation of the technologies involved in web development and how the internet works. Once the introduction is done, HTML and CSS should be introduced, making as many possible examples and exercises until the students feel comfortable understanding the structure of a webpage using both languages. For this, you could start with the Task Sheet D1.1 (create your first web page) or Task sheet D1.2 (Create your first webpage using a visual block editor) as a good introduction and to show the difference between using a framework and using coding languages to build the same website. Task sheet D1.3 (Find the errors) will help students to understand the code in a playful manner, and Task sheet D1.4 (creating a website based on a specific target group (including design thinking)) includes many coding challenges that can also be added with new ones to get students even more used to coding with HTML and CSS.

After learning the basics of HTML and CSS, JavaScript is the perfect way to improve the quality of a webpage. In order to show the basics of JavaScript to the students, you can use the Task sheet D1.5 (make your web page interactive with javascript) and Task sheet D1.6 (create your first interactive e-commerce).

Lastly, Task sheet D1.7 (create a working contact form) can be used as an introduction to PHP and also to explain the differences between a frontend and backend language.

This handout can be offered together with microcontrollers to arrive at Internet of Things in which microcontrollers are combined with a web page.

Additional information:

- Article on web development: https://en.wikipedia.org/wiki/Web_development
- Web development Learn Path: https://www.w3schools.com/whatis/

Module: Web development

Topic: Wordpress

General Description:

Web programming, also known as web development, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

For the creation of web pages, you can use programming languages (HTML, CSS, JAVA, etc.) or you can use a framework (Wordpress). This handout focuses on how to use the Wordpress Framework in combination with some coding language (variables and CSS) and explores its possibilities for web pages, blogs and online shops (e-commerce).

Learning objectives:

Handout learning objective:

- Understand how coding works
- Have basic understanding how to create posts and pages
- Know how to connect wordpress on wamp/xampp server

Digital skills learning objective:

- 1.1
- 2.1; 2.2
- 3.1; 3.2; 3.4
- 5.2; 5.3

Preparatory steps for the trainer:

- Learn how to create posts and pages
- Learn how to install wordpress in local server

Exercises:

- Task sheet D2.1 – Connect wordpress on wamp server
- Task sheet D2.2 – Create pages and posts on wordpress

How to organise your learning:

You can start introducing the topic to the students by giving a brief explanation of web pages and e-shops and then, continue with the basic coding languages and the basics of wordpress. The ideal would be to think of one or two topics each week, and let students try out their own wordpress pages using the new items.

You could start with a theoretical introduction to coding and wordpress. Then use Task Sheet D2.1 (connect wordpress on wamp server) to practice some of the basic necessary information. This could be followed by working on wordpress, through Task Sheet D2.2 (create pages and posts on wordpress).

Once the basics are set, you can start to motivate them to think on their own projects and what kind of webpage or web shop they would like to develop.

This handout can be offered together with Basic Web Development

- Handout: Basic Web Development
  - The basic web development handout will focus on introducing students to the coding language and creating web pages using coding language without a framework. Also understanding basic HTML and CSS code can be helpful in the creation of Wordpress content.

Additional information:

- More information about Wordpress: https://www.w3schools.in/wordpress-tutorial/intro/

- Themes:
  - https://wordpress.org/themes/calliope/
  - https://wordpress.org/themes/newsup/

- Plugins:
  - https://wordpress.org/plugins/advanced-posts-blocks/
  - https://wordpress.org/plugins/contact-form-7/
  - https://wordpress.org/plugins/wordpress-seo/
  - https://wordpress.org/plugins/woocommerce/