Co-funded by the European Union

Tutorial 1: Understanding SmartHome

1. Introduction

In this tutorial, you will learn about the SmartHome Kit, how to assemble it, what the Raspberry Pi Pico is and how the programming interface works. Some of this information is described in the SmartHome manual, however with this introductory tutorial, you will have the chance to watch educational video tutorials on how the SmartHome is assembled and how the microcontroller should be initiated.

1.1. Learning Objectives

- Understand the concept of Smart Home and assembly the SmartHome Kit.
- Understand the Raspberry Pi Pico microcontroller and its programming environment.
- Use the print function.

1.2. Theoretical background

The SmartHome4SENIORS Kit is constructed using the Raspberry Pi Pico (RPi Pico) microcontroller as its foundation and several plywood pieces that simulate a house.

Smart home automation is a growing trend that simplifies household processes for many people. The conception and development of the SmartHome4SENIORS Kit was driven by the recognition that seniors often have a limited familiarity with and understanding of smart home technology.

The primary goal of the Kit is to encourage safe and healthy living among seniors through hands-on educational experiences. It provides an excellent opportunity for users to explore the world of do-it-yourself (DIY) smart home automation solutions.

The Kit comes complete with all the essential hardware components, including the microcontroller, electronics, sensors, peripherals, and more, which facilitate the understanding of physical computing and programming principles.

1.3. Material

For this tutorial you will need the following material:

- Plywood pieces (x6)
- Raspberry Pi Pico
- Breadboard
- MB-102 power supply unit
- Jumper wires

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

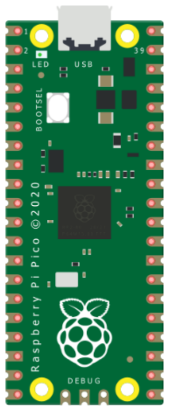
2. Initial steps & Connectivity

1. Readme Before Using

NOTE: Since the experiments involved are all circuit experiments, a wrong connection or short circuit may damage your RPi Pico development board. Please, always check the circuit again before connecting the power supply.

2. Raspberry Pi Pico Microcontroller

This is the Raspberry Pi Pico:



The Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation, known for its popular Raspberry Pi single-board computers (SBCs). Unlike the full-fledged Raspberry Pi SBCs, the Raspberry Pi Pico is designed for microcontroller and embedded systems projects.

Key features of the Raspberry Pi Pico include:

1. **RP2040 Microcontroller:** The Pico is powered by the RP2040 microcontroller chip, which was also developed by the Raspberry Pi Foundation. The RP2040 is a dual-core ARM Cortex-M0+ microcontroller with a clock speed of up to 133MHz.
2. **Connectivity:** It features a variety of GPIO (General-Purpose Input/Output) pins that can be used for digital input/output, analog input, PWM (Pulse-Width Modulation), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), and UART (Universal Asynchronous Receiver-Transmitter) communication.
3. **USB Connectivity:** The Pico can be powered and programmed via its micro-USB port, making it easy to connect to a computer for programming and power supply.
4. **Programmability:** It can be programmed using a variety of programming languages, including MicroPython, C/C++, and CircuitPython, which makes it accessible to both beginners and experienced developers.
5. **Low-Cost:** The Raspberry Pi Pico is affordable, making it an excellent choice for educational and hobbyist projects.

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

6. **Community Support:** The Raspberry Pi Foundation provides extensive documentation, tutorials, and a supportive community, making it easier for users to get started with their projects.

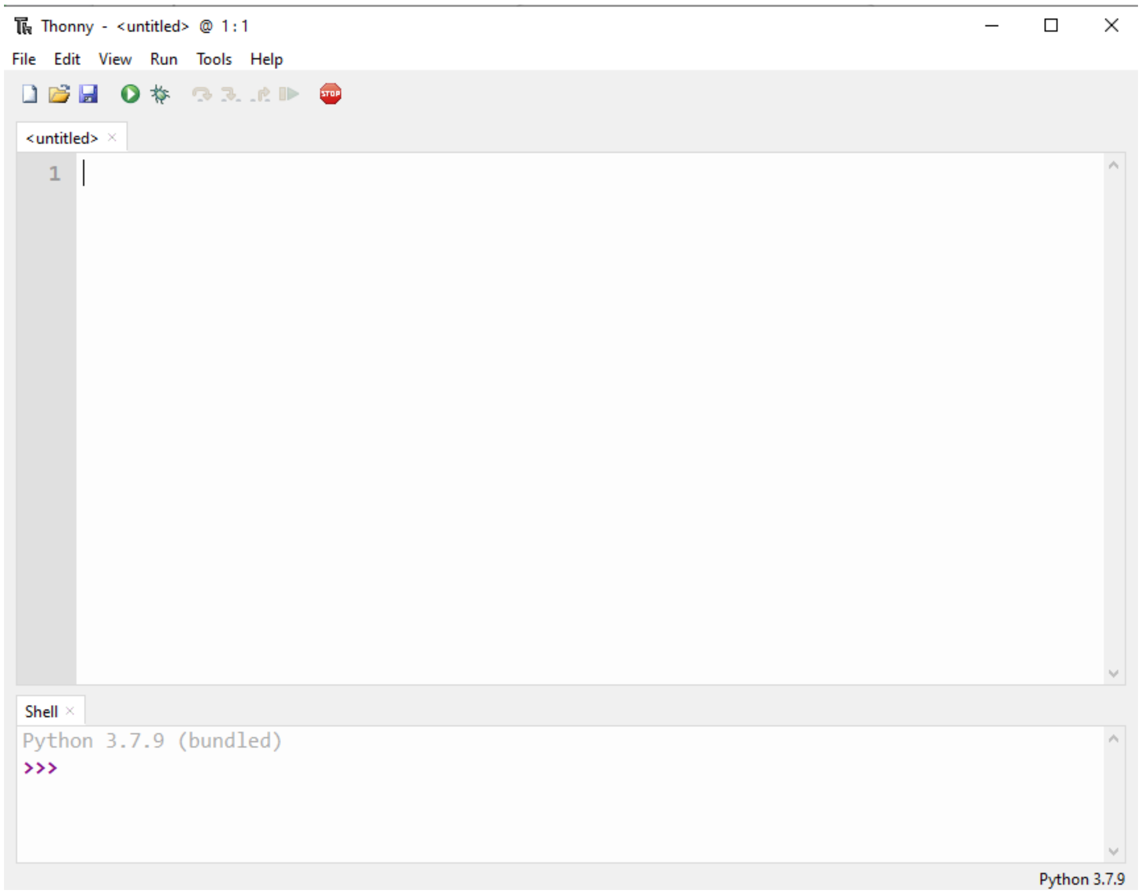
The Raspberry Pi Pico is suitable for a wide range of applications, including robotics, home automation, IoT (Internet of Things) devices, sensors, and more. It has gained popularity for its versatility and affordability, making it an attractive option for makers and developers looking to create embedded systems and microcontroller-based projects.

3. Install Thonny IDE

Visit https://thonny.org and choose the appropriate operating system. Follow the instructions to complete the installation.

In this manual, all tutorials are programmed in Windows 10, using a RPi Pico microcontroller and the appropriate firmware.

After installation is completed, open Thonny from your computer.



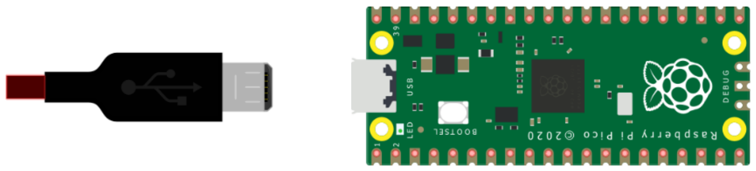
```
Python 3.7.9 (bundled)
>>>
```

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

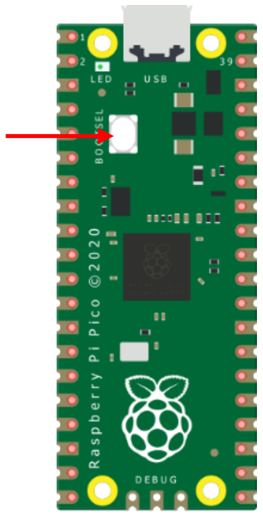
4. Firmware installation

The RPi Pico can be programmed using a Python variant, called MicroPython. To use MicroPython on the RPi Pico, first you need to install its firmware.

Step 1: Plug the micro-USB cable into the port on the left-hand side of the board.

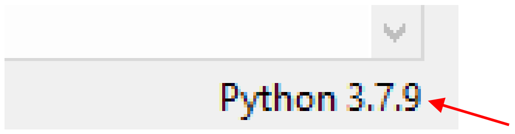


Step 2: Find the BOOTSEL button on your Raspberry Pi Pico.

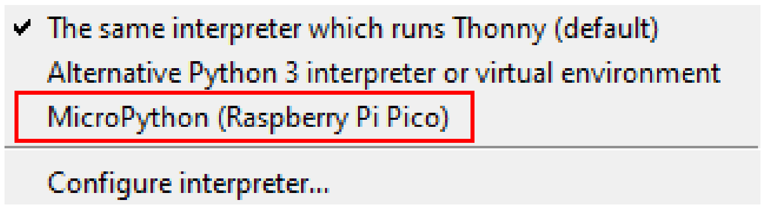


Step 3: Press the BOOTSEL button and hold it while you connect the other end of the micro-USB cable to your computer.

Step 4: In the bottom right-hand corner of Thonny you will see the version of Python you currently use.

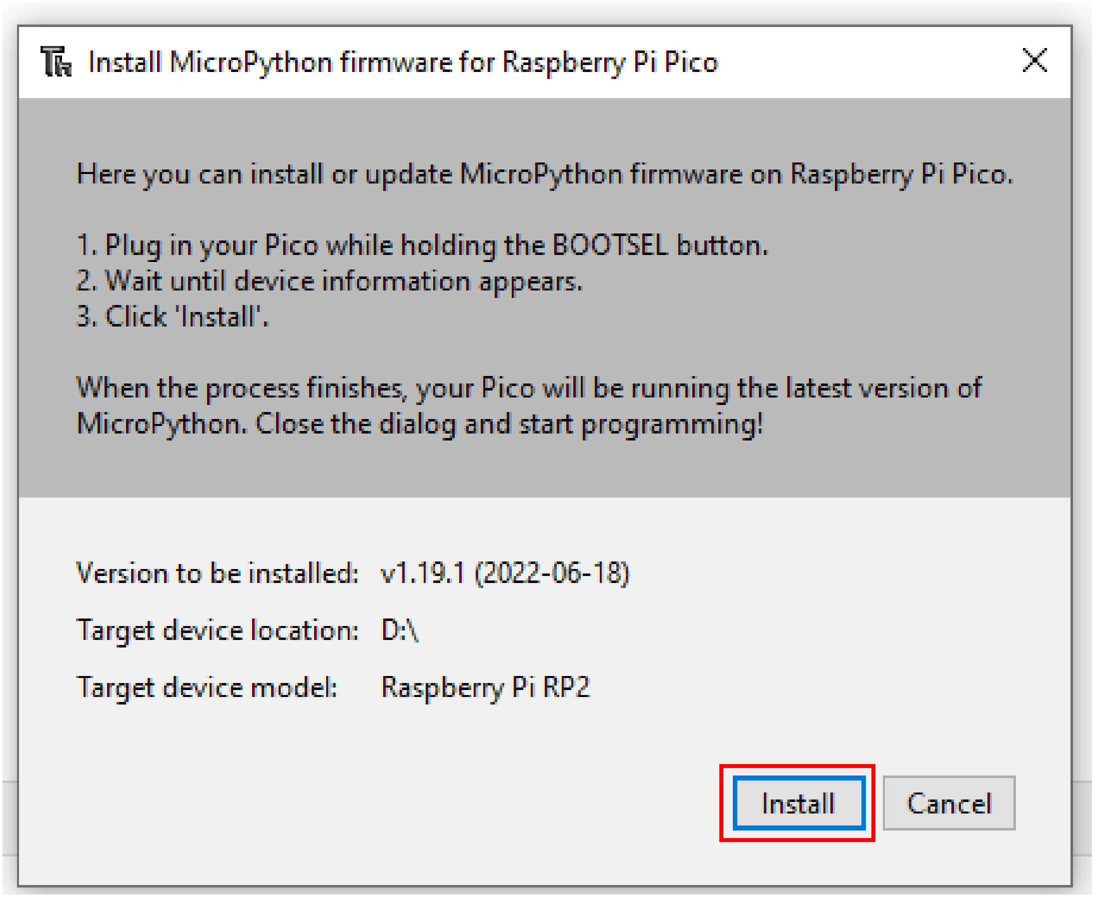


Click on the Python version and choose the MicroPython (Raspberry Pi Pico)


✓ The same interpreter which runs Thonny (default)
Alternative Python 3 interpreter or virtual environment
MicroPython (Raspberry Pi Pico)
Configure interpreter...

If you don't see this option, make sure the cable is connected properly on the Pico and/or your computer.

Step 5: A dialog box will appear, asking you to install the latest firmware version to your Pico. Click the **Install** button to copy the firmware to your Pico.



Install MicroPython firmware for Raspberry Pi Pico

Here you can install or update MicroPython firmware on Raspberry Pi Pico.

1. Plug in your Pico while holding the BOOTSEL button.
2. Wait until device information appears.
3. Click 'Install'.

When the process finishes, your Pico will be running the latest version of MicroPython. Close the dialog and start programming!

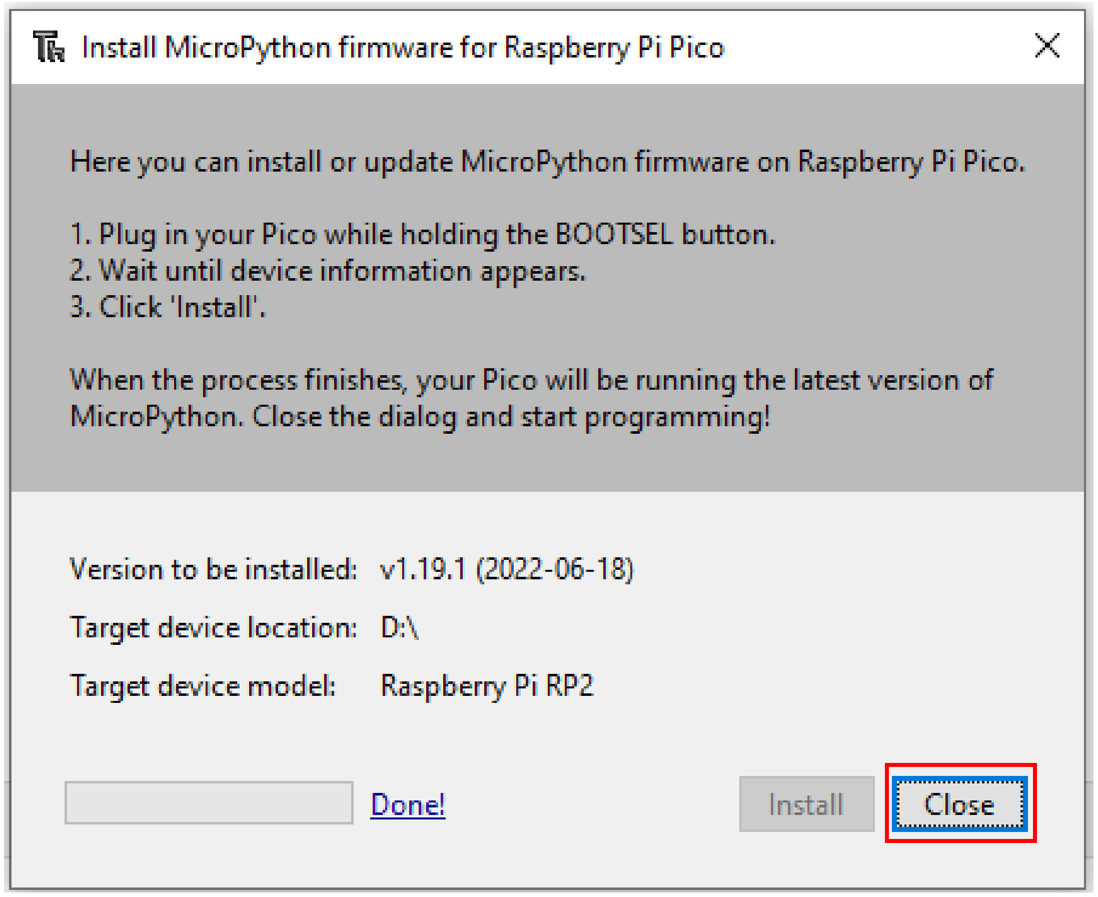
Version to be installed: v1.19.1 (2022-06-18)

Target device location: D:\

Target device model: Raspberry Pi RP2

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

Step 6: Wait for the installation to complete and click **Close**.



Install MicroPython firmware for Raspberry Pi Pico

Here you can install or update MicroPython firmware on Raspberry Pi Pico.

1. Plug in your Pico while holding the BOOTSEL button.
2. Wait until device information appears.
3. Click 'Install'.

When the process finishes, your Pico will be running the latest version of MicroPython. Close the dialog and start programming!

Version to be installed: v1.19.1 (2022-06-18)

Target device location: D:\

Target device model: Raspberry Pi RP2

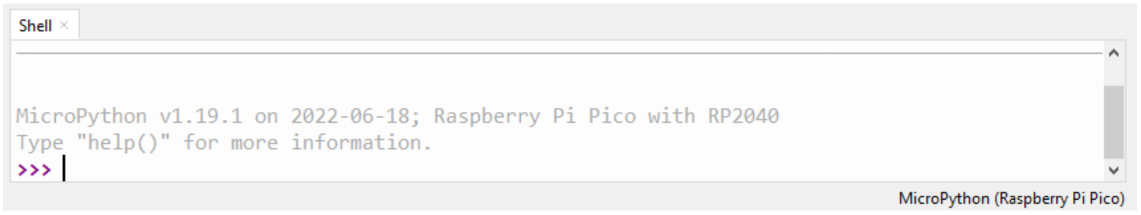
You don't need to repeat the process every time you connect the Raspberry Pi Pico to your computer, so next time, just plug it in and you are good to go.

5. Introduction to MicroPython programming

You will now use Thonny IDE to run some simple Python code to get acquainted with Thonny's Shell and MicroPython.

First make sure that your Raspberry Pi Pico is connected to your computer, and you have selected the MicroPython interpreter as explained in the previous section.

The Shell panel at the bottom of Thonny editor should look like this:



```
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

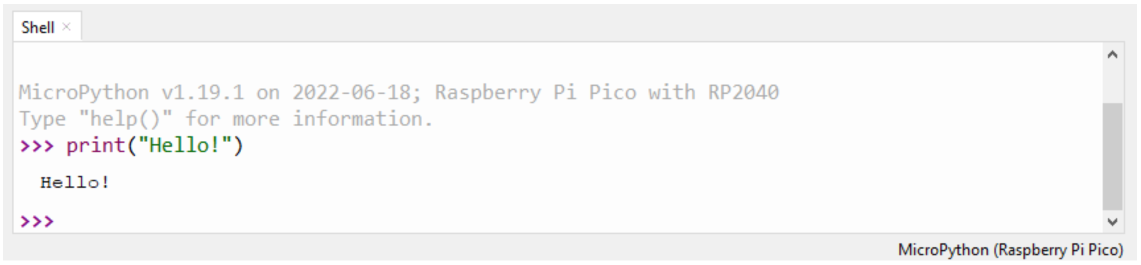
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Thonny is ready to communicate with your Pico using REPL (read-eval-print loop) framework, which allows you to write code directly at the Shell and get output.

Type the following command:

```
print(“Hello!”)
```

Then hit the Enter key and see the following output:

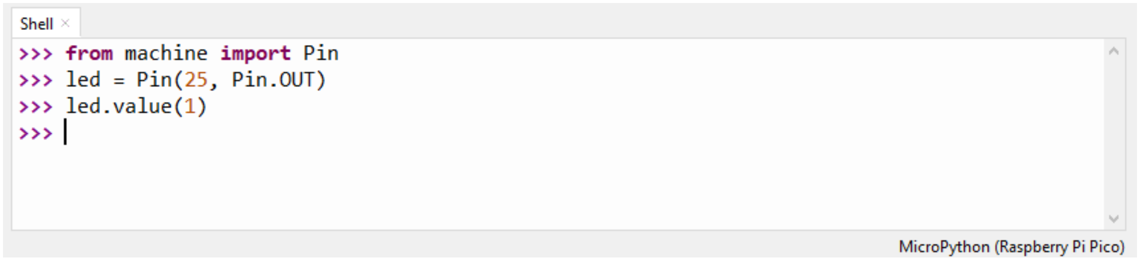


```
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> print("Hello!")
  Hello!
>>>
```

MicroPython allows you to add hardware-specific modules, such as `machine`, that you can use to program your Pico. In the following example you will use the `machine` module to turn on Pico’s onboard LED.

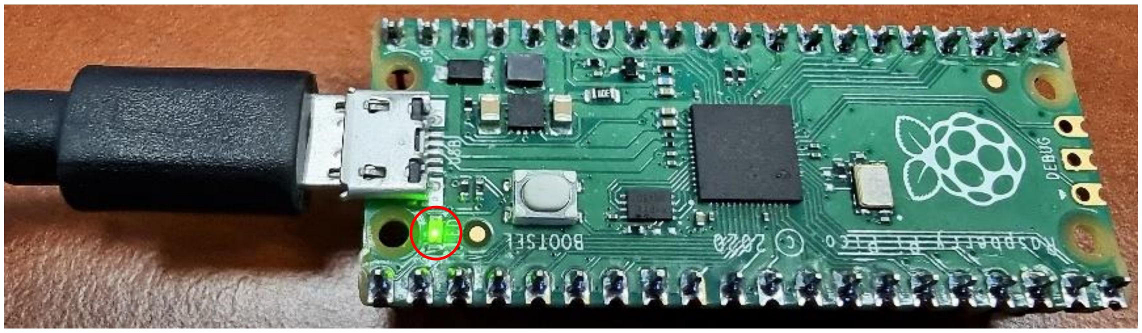
Write the following code in Thonny’s Shell:

```
from machine import Pin
led = Pin(25, Pin.OUT)
led.value(1)
```



```
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>>
```

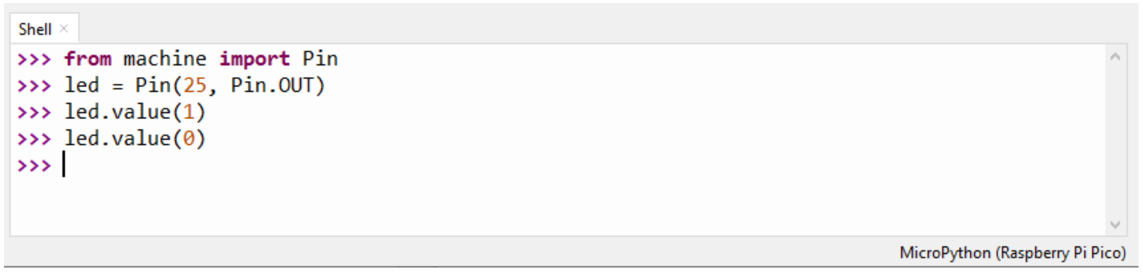
Press the Enter key and immediately the Pico’s onboard LED will turn on.



To turn off the LED write the following code:

```
led.value(0)
```

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



```
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> led.value(0)
>>>
```

For the rest of this section, you will write your first "real" program that will make the onboard LED to blink every time you run your program.

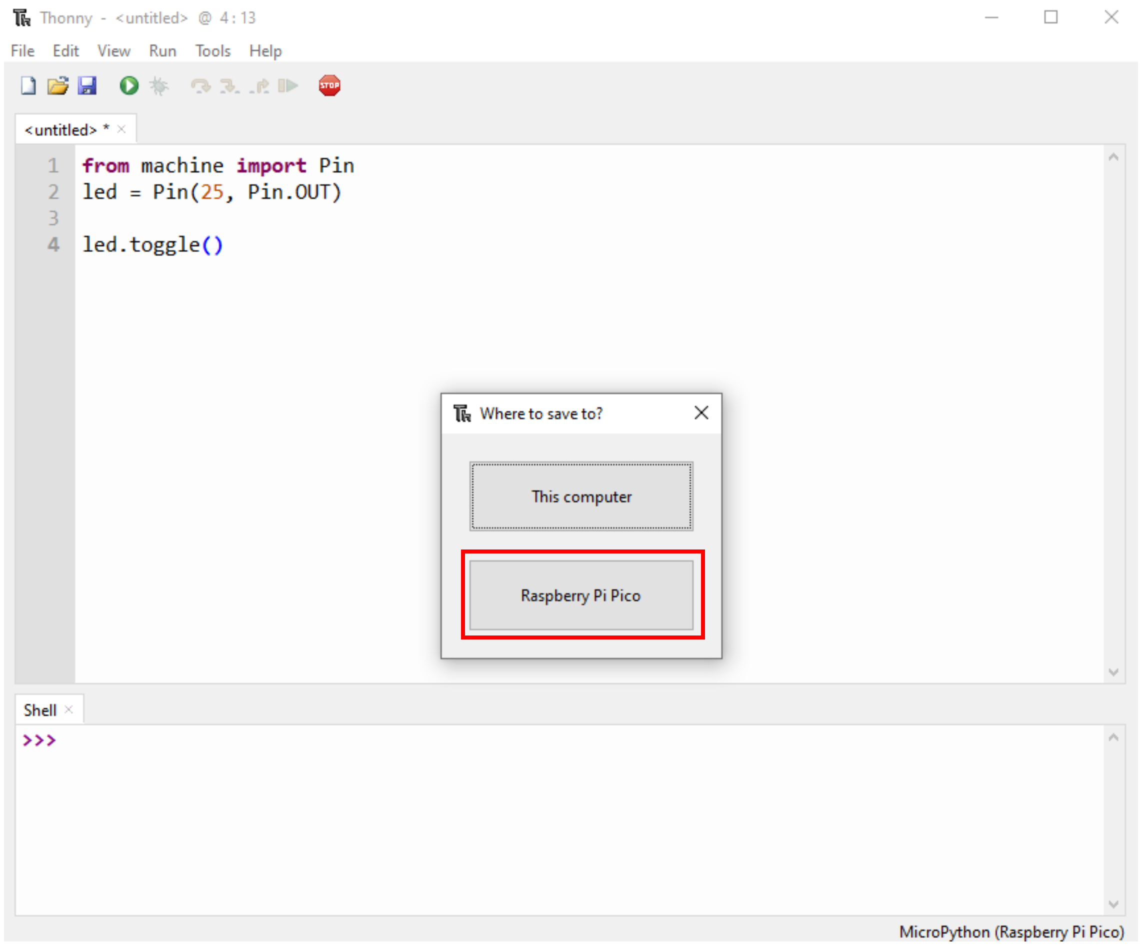
Thonny Shell is useful to try out quick commands and make sure everything is working correctly. However, longer programs should be saved in a `.py` file. Using Thonny, you can save programs directly to the Raspberry Pi Pico and then run them.

Open Thonny Python and on the main editor pane write the following code:

```
from machine import Pin
led = Pin(25, Pin.OUT)
led.toggle()
```

Now, save your program by clicking the Save icon on the top left-hand side, or by pressing Ctrl+S on your keyboard.

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

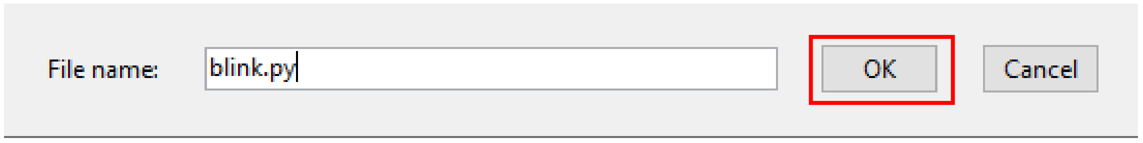

Thonny - <untitled> @ 4 : 13
File Edit View Run Tools Help
<untitled> *
Where to save to?
This computer
Raspberry Pi Pico
Shell
MicroPython (Raspberry Pi Pico)

```
from machine import Pin
led = Pin(25, Pin.OUT)

led.toggle()
```

```
>>>
```

Thonny will ask you where you want your program to be saved. Choose **Raspberry Pi Pico**. Save the file as *blink.py* and click **OK**. You always need to add the .py extension so that Thonny recognises the file as a Python file.



Now, every time you click the Play icon, you should see the onboard LED switching ON and OFF.

Taking your code one step further, you can make the onboard LED blinking at a certain pace.

Write the following code and save your program using the same name as above.

```
from machine import Pin
from time import sleep
led = Pin(25, Pin.OUT)
while True:
      led.toggle()
```

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

```
        sleep(1)
```

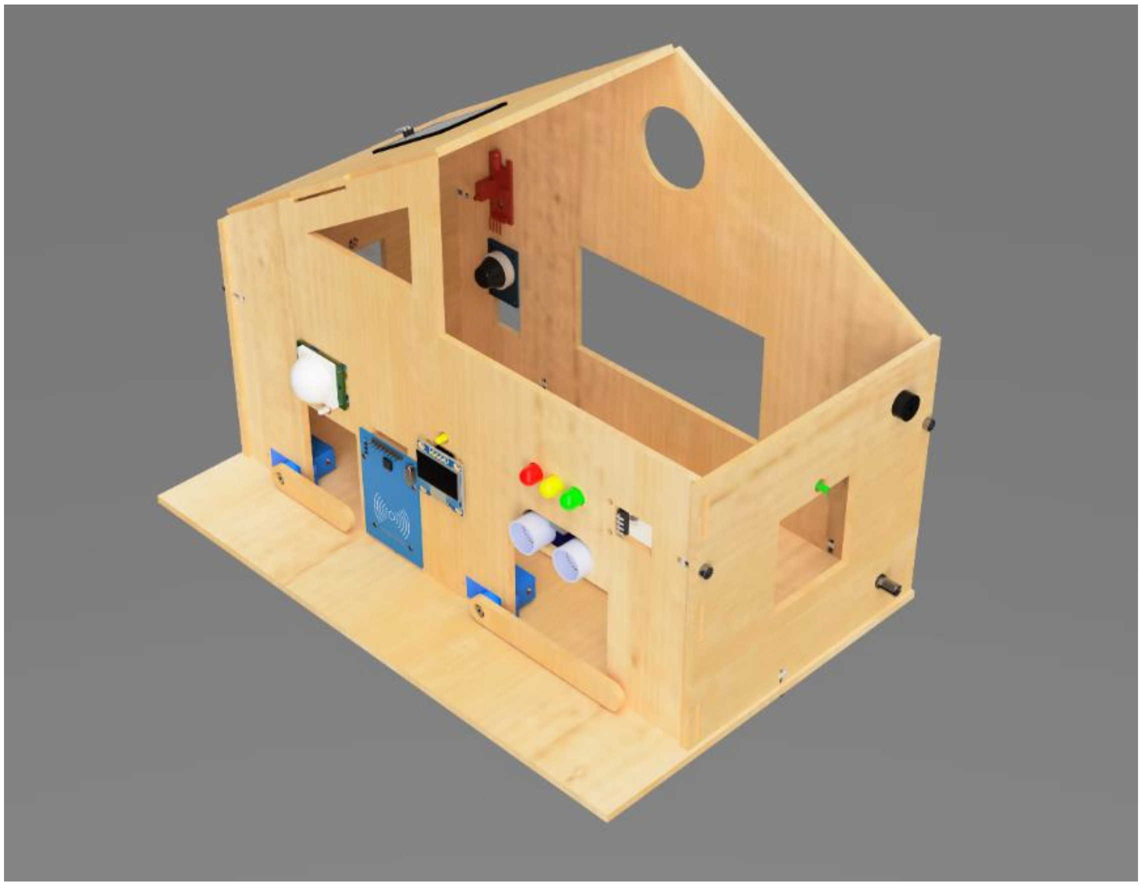
Now when you run the program, the onboard LED will blink every 1 second until we stop the program. To stop the program, you can click on the STOP icon or press Ctrl+C on your keyboard.

In future tutorials, we will learn how to add and control other electronics and sensors and create programs that can communicate with each other.

6. Assembly the SmartHome

The assembly process requires the following items:

- 6 x Plywood pieces
- 10 x Metal bolts
- 10 x Metal nuts



The procedure is simple and can be completed in **7 steps**. All plywood pieces are marked on the right-hand side corner with the following notation:

BP: Base piece

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

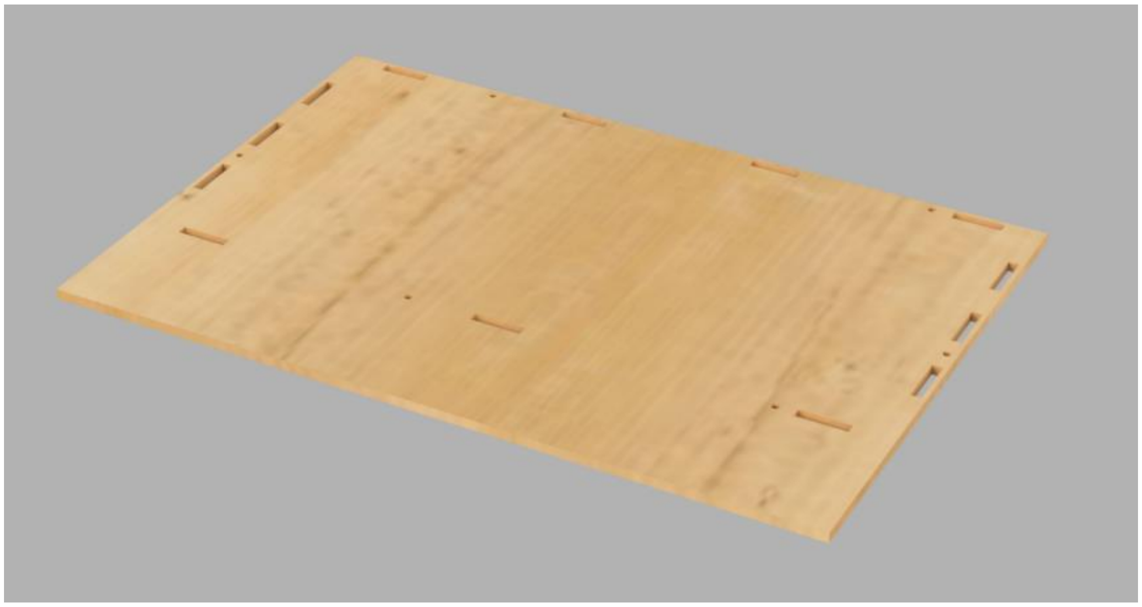
B: Back-side piece

F: Front-side piece

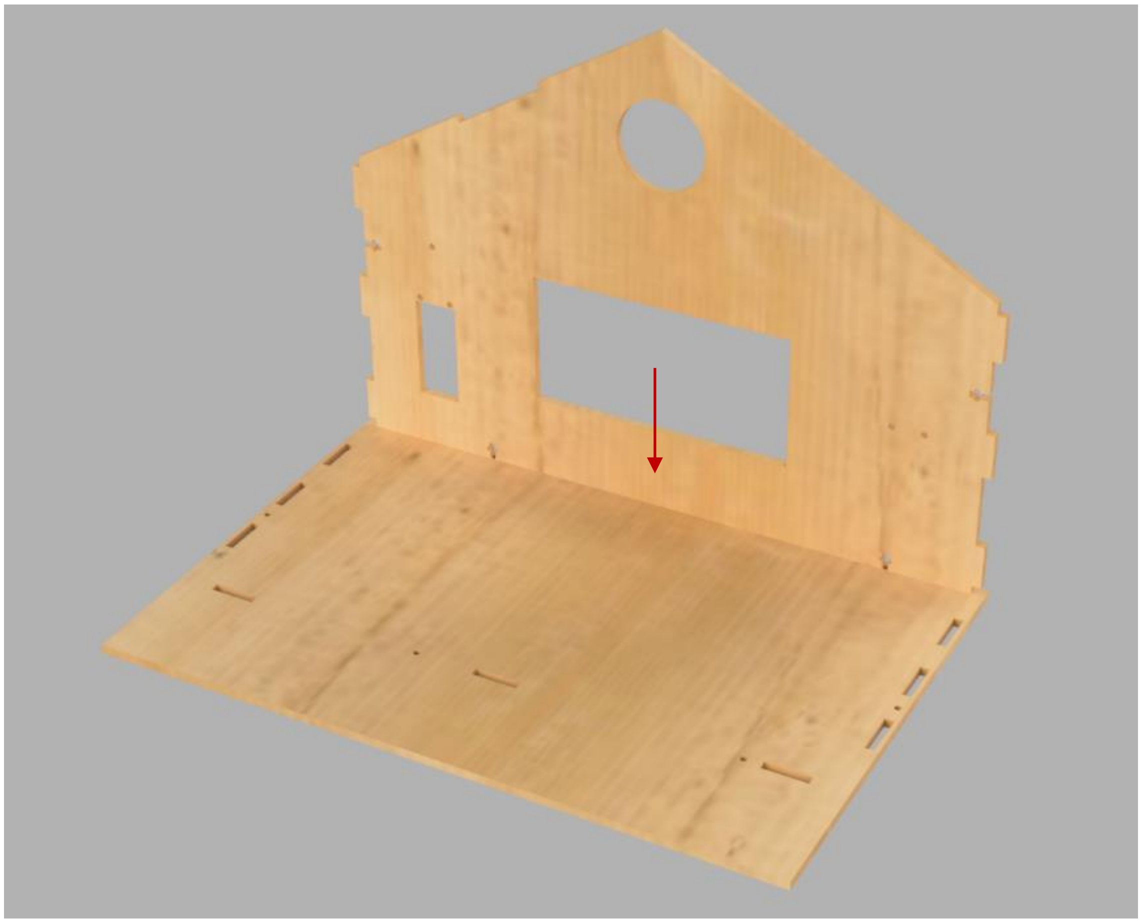
L: Left-side piece

R: Right-side piece

Step 1: Place the plywood base piece on a horizontal surface.

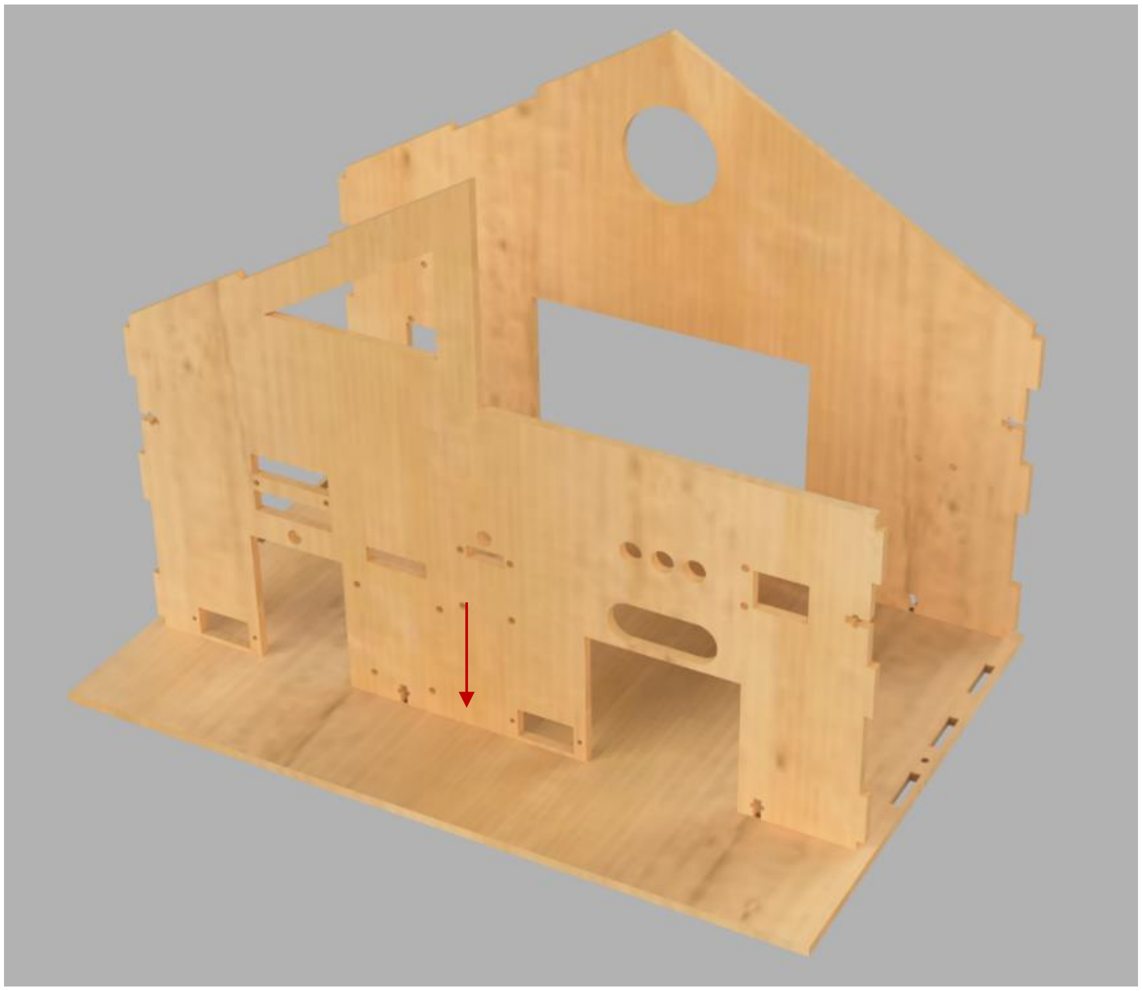


Step 2: Insert the back-side piece as shown in the image.



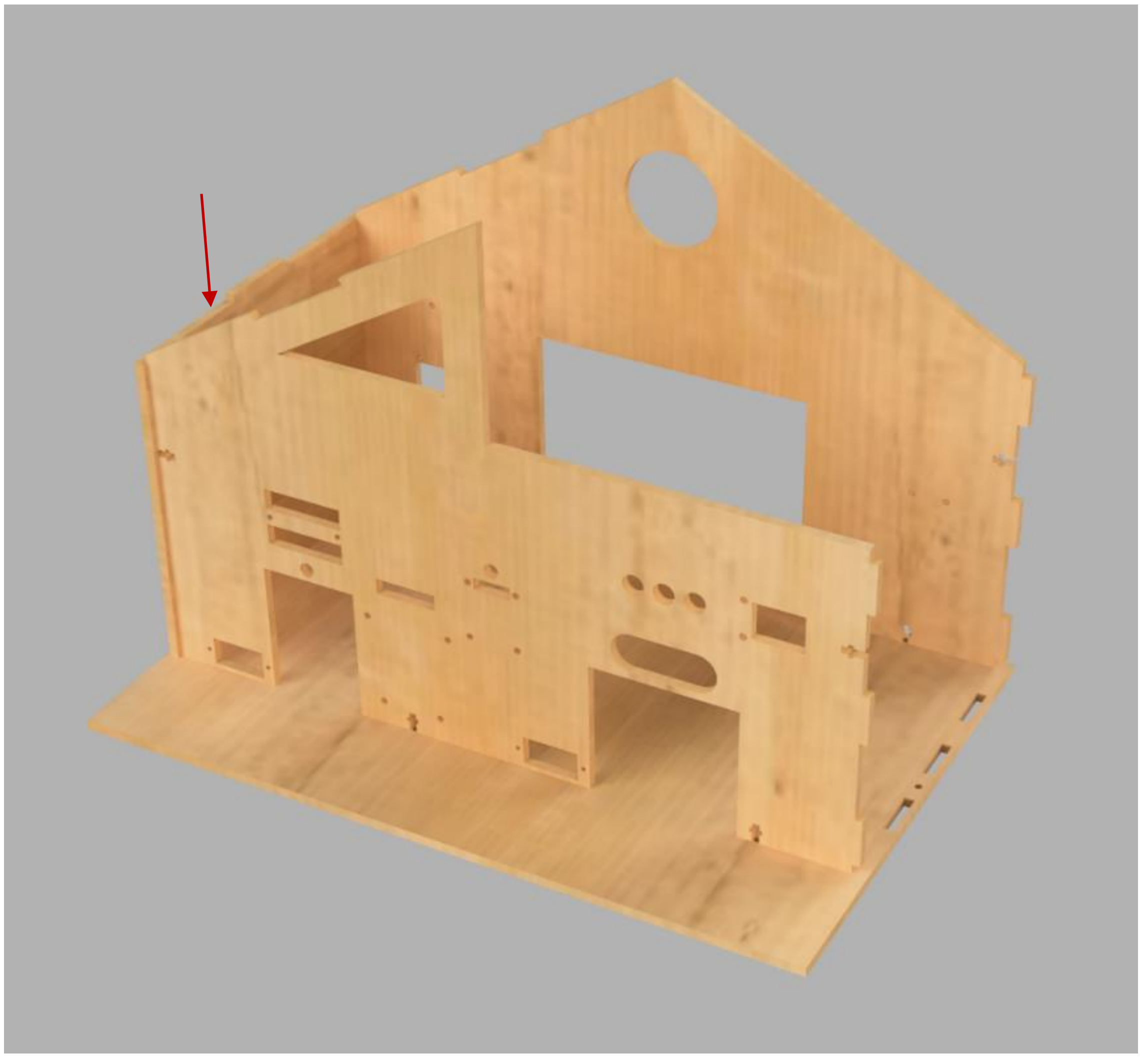
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 3: Insert the front-side piece as shown in the image.



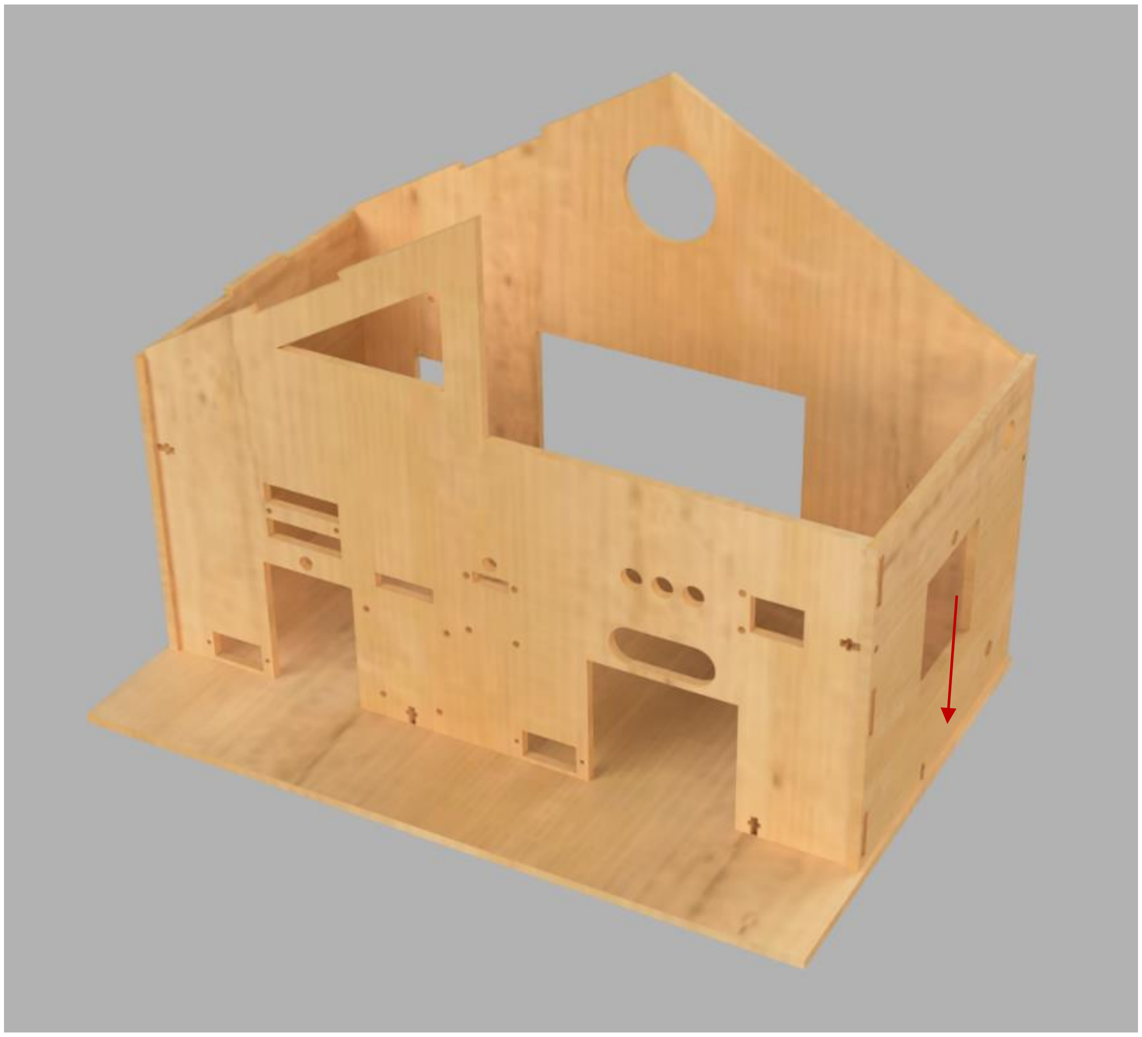
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 4: Insert the plywood left-side piece as shown in the image.



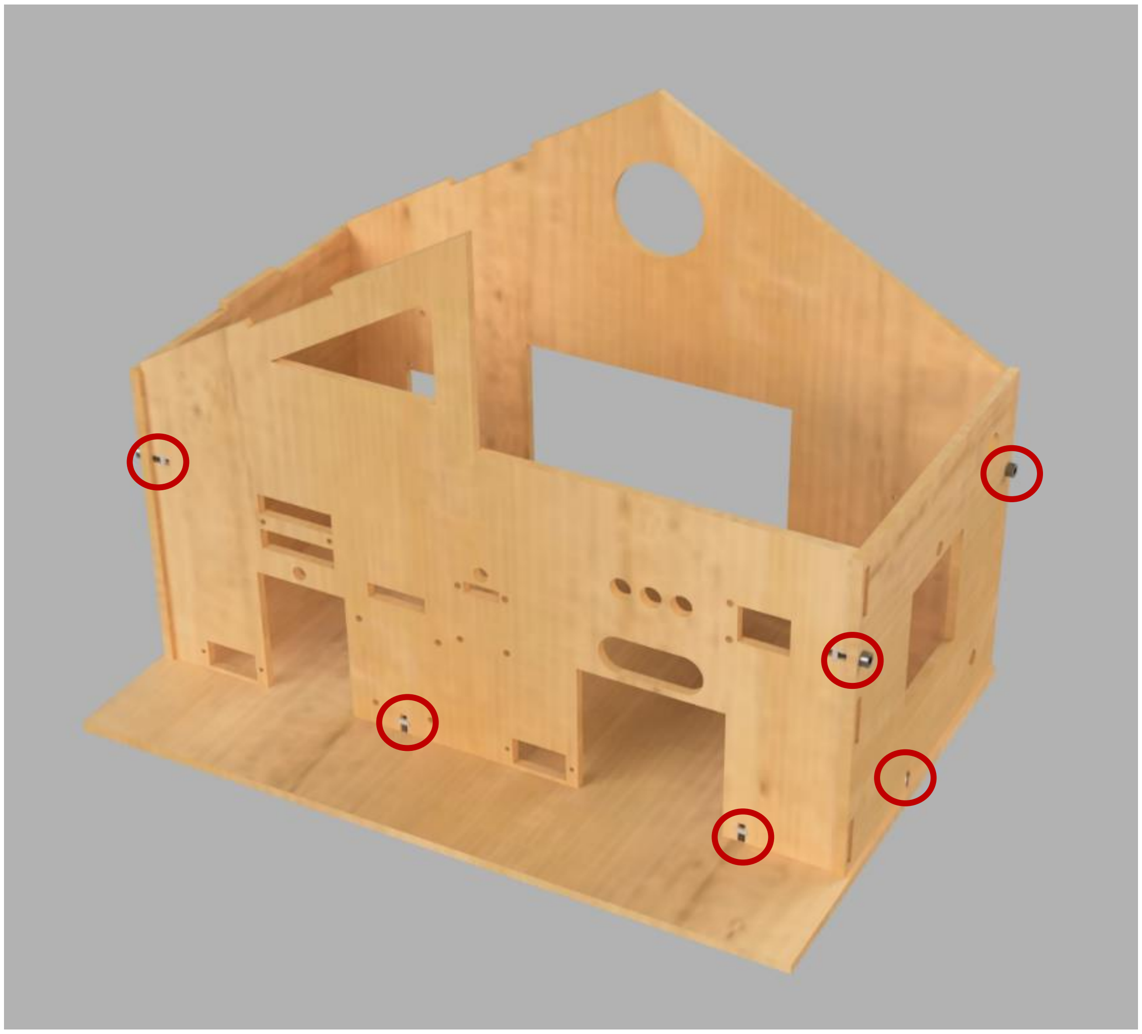
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 5: Insert the plywood right-side piece as shown in the image.



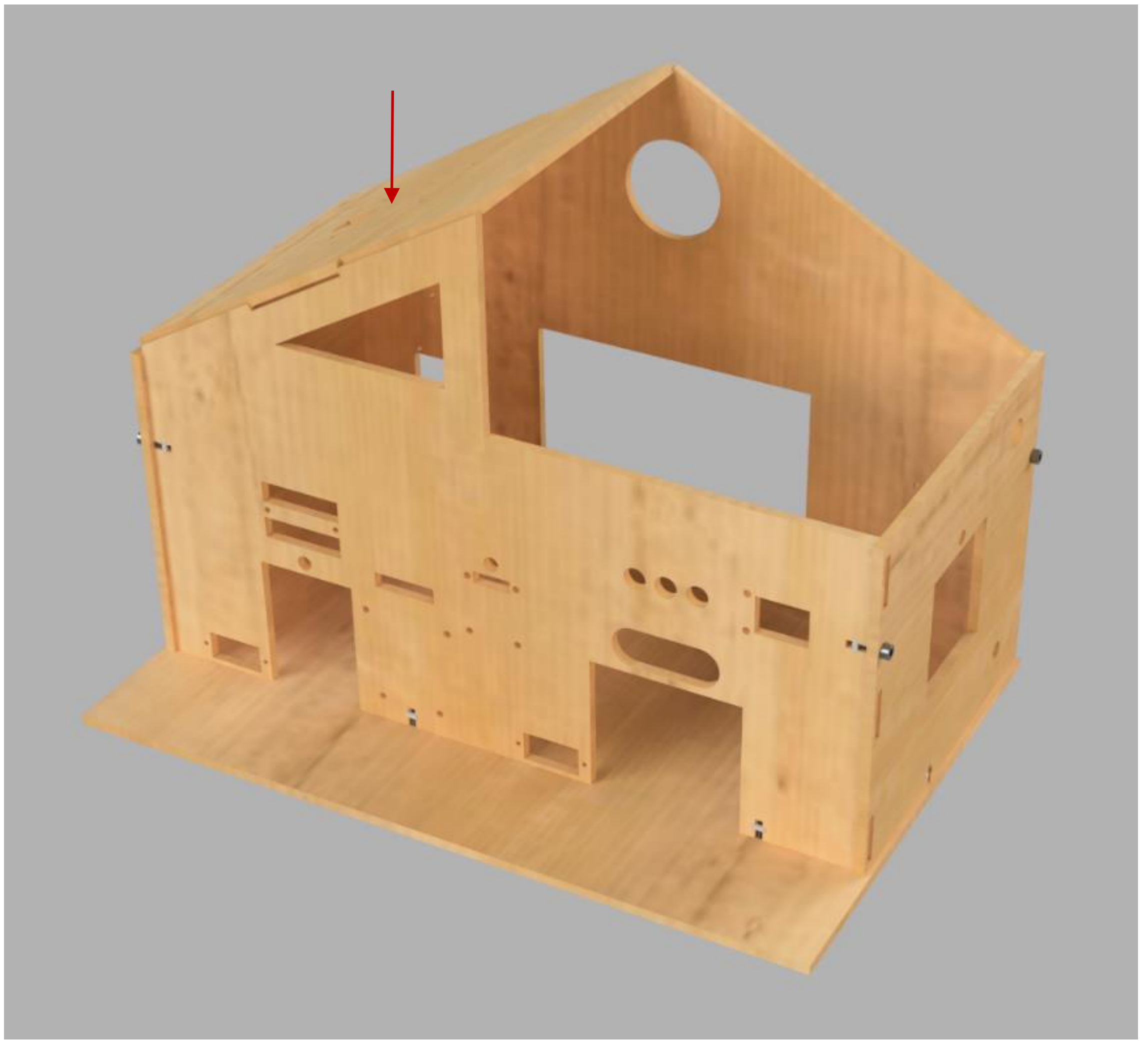
Step 6: Tighten the plywood pieces together using 10 bolts and 10 nuts.

- 6 bolts and nuts are placed under the base piece and are responsible for keeping together the walls.
- 4 bolts and nuts are placed on the left-side and right-side walls and are responsible for keeping the construction stable.



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

Step 7: Place the plywood roof piece on top of the house model as shown in the image:



7. Setting up SmartHome for upcoming tutorials

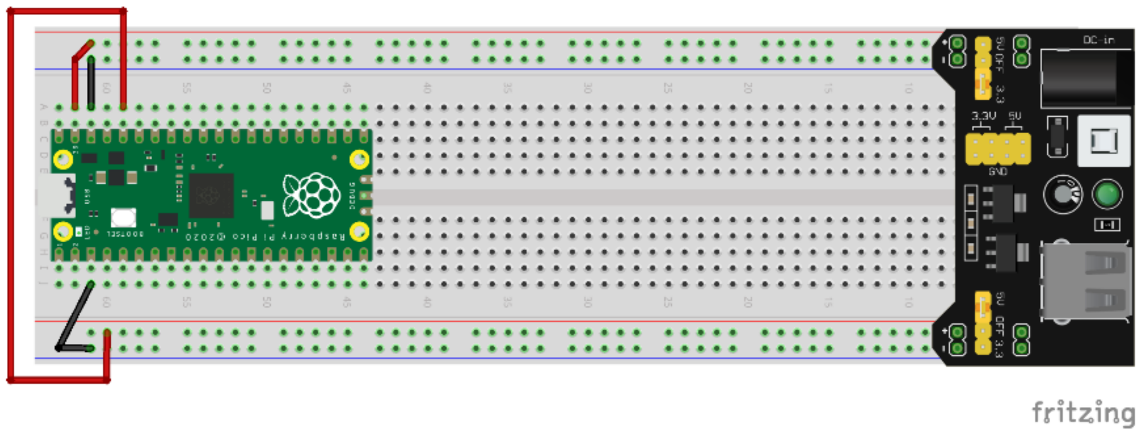
You need to make a few modifications to your development board before proceeding with the other tutorials. This involves the addition of a few extra components and their connectivity.

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

Components

- 1 x 6-AA battery pack
- 1 x MB-102 power module
- 4 x Male-to-male jumper wires

Please follow the schematic to connect the new components:

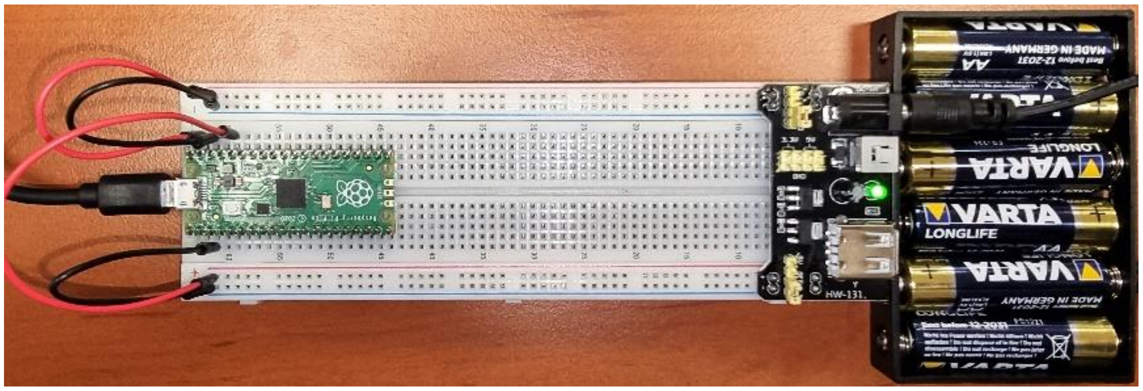


– this setup will be used for the remaining tutorials

– top side connections: VSYS 5V ((+) red) and GND ((-) black)

– bottom side connections: 3V3 ((+) red) and GND ((-) black)

Your setup should look like the following picture:



Then you need to attach the breadboard to the SmartHome floor, similar to the following picture:

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

8. Summary

In this introductory tutorial you have learned about the Raspberry Pi Pico microcontroller and how to install the necessary software and firmware before using it. You have also learned how to assemble the SmartHome wooden model, and to prepare it for the upcoming tutorials. The setup of this tutorial included the installation of the following components:

1. Raspberry Pi Pico
2. Breadboard
3. MB-102 power supply module
4. Plywood pieces
5. Metal bolts and nuts
6. Jumper wires
7. USB to micro-USB cable

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