

## LAFVIN Pico

# LAFVIN Basic Starter Kit for Raspberry Pi Pico: Instruction Manual

Model: Pico | Brand: LAFVIN

## 1. INTRODUCTION

---

The LAFVIN Basic Starter Kit for Raspberry Pi Pico is designed to introduce users to the fundamentals of electronics and programming. This kit provides a variety of components and comprehensive tutorials to facilitate learning with the Raspberry Pi Pico microcontroller.

## 2. WHAT'S IN THE BOX

---

The kit includes the following components:

- 1 x Raspberry Pi Pico
- 1 x IIC LCD 1602 Display Module
- 1 x 830 Tie-Points Breadboard
- 1 x Obstacle Avoidance Module
- 1 x HC-SR501 PIR Motion Sensor
- 1 x SG90 Servo
- 1 x Potentiometer (10k)
- 1 x Passive Buzzer
- 1 x Active Buzzer
- 30 x Resistors (10x 220R, 10x 1k, 10x 10k)
- 1 x Micro-USB Cable
- 8 x Button Switches
- 1 x WS2812 RGB 16-LED Ring Module
- 10 x F-M DuPont Cables
- 65 x Jumper Wires
- 1 x Thermistor
- 1 x Tilt Switch

- 2 x Photoresistors
- 2 x S8050 Transistors
- 5 x Red LEDs
- 5 x Yellow LEDs
- 5 x Green LEDs
- 1 x RGB LED



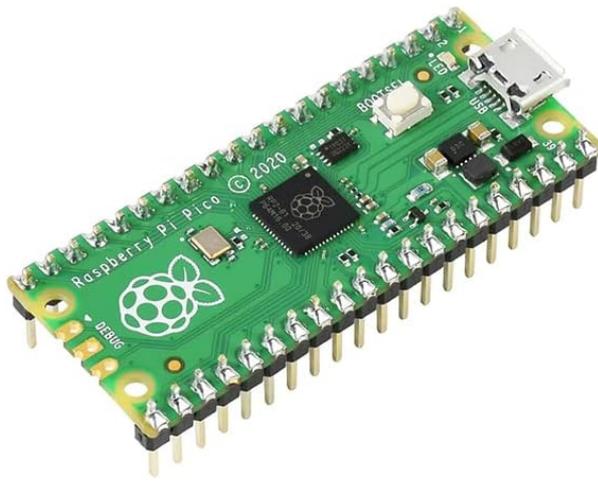
Figure 2.1: All components included in the LAFVIN Basic Starter Kit.

This image displays all the individual components you will find in your LAFVIN Basic Starter Kit, neatly organized and labeled for easy identification.

### 3. KEY FEATURES

---

- The Raspberry Pi Pico series is a range of tiny, fast, and versatile boards built using RP2040.
- Piper Make allows users to learn and absorb the basics of programming in the electronics industry as they learn how to build their own computers.
- The tutorial includes codes and lessons. It will teach every user how to assemble the Basic Starter Kit for Pi Pico.
- Supports 3 languages: MicroPython, C, and Piper Make.



## Key Features:

- RP2040 microcontroller with 2MB Flash
- Micro-USB B port for power and data (and for reprogramming the Flash)
- 40 pin 21x51 'DIP' style 1mm thick PCB with 0.1" through-hole pins also with edge castellations

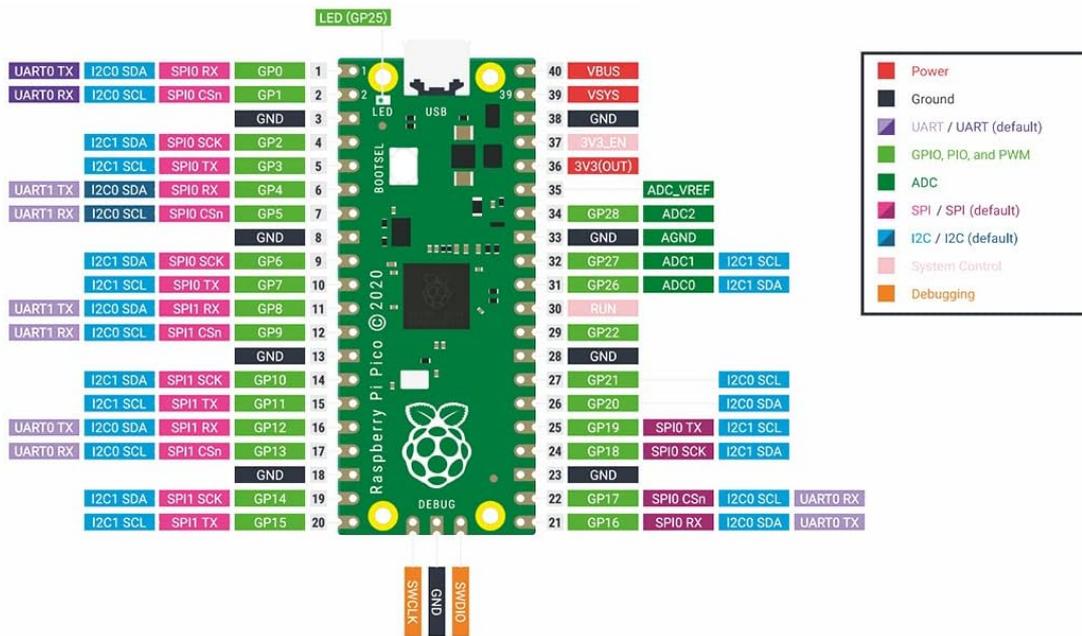


Figure 3.1: Raspberry Pi Pico board with key features.

This image provides a close-up view of the Raspberry Pi Pico board, detailing its RP2040 microcontroller and Micro-USB B port for power and data.

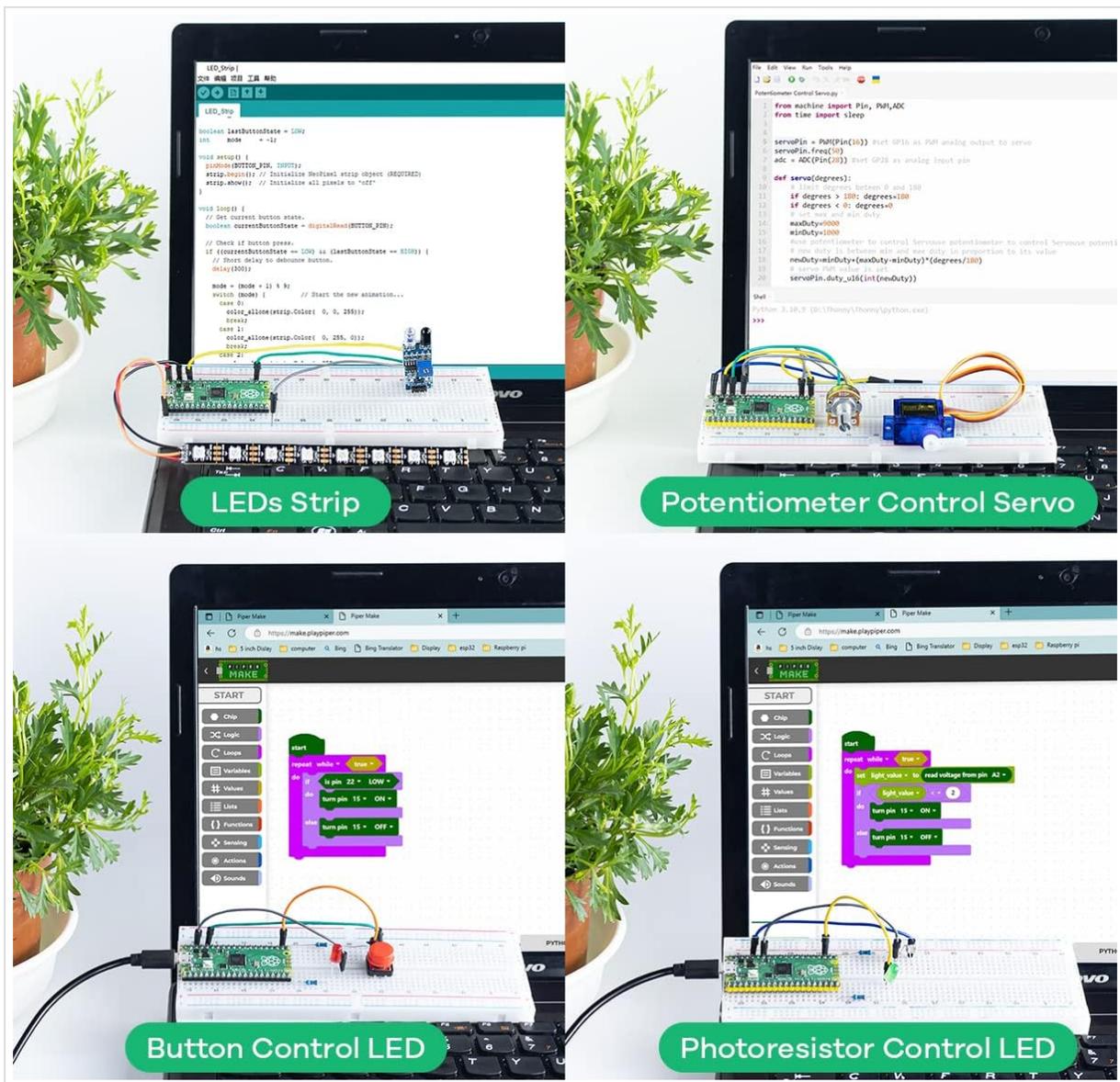


Figure 3.2: Raspberry Pi Pico pinout diagram.

This diagram illustrates the pinout of the Raspberry Pi Pico, showing the arrangement and function of its General Purpose Input/Output (GPIO) pins, power pins, and communication interfaces.

## 4. SETUP AND GETTING STARTED

To begin using your LAFVIN Basic Starter Kit, follow these general steps:

1. **Unpack Components:** Carefully remove all components from the packaging and verify against the 'What's in the Box' list.
2. **Review Documentation:** Familiarize yourself with the provided tutorial and online resources.
3. **Connect Raspberry Pi Pico:** Insert the Raspberry Pi Pico into the breadboard. Ensure all pins are correctly aligned.
4. **Assemble Circuits:** Follow the step-by-step instructions in the tutorial to build your first circuit using the breadboard, jumper wires, and other components.
5. **Connect to Computer:** Use the Micro-USB cable to connect your Raspberry Pi Pico to your computer.
6. **Install Software:** Install the necessary software (e.g., Thonny IDE) and MicroPython firmware as guided by the tutorial.
7. **Upload Code:** Upload the provided example code or write your own to the Raspberry Pi Pico.

Your browser does not support the video tag.

Video 4.1: Product Use Tutorial by HONGSHUYUAN.

This video provides a general overview and demonstration of how to use the LAFVIN Basic Starter Kit, including unboxing and initial setup steps.

Your browser does not support the video tag.

Video 4.2: Basic Starter Kit Overview by HONGSHUYUAN.

This video offers another perspective on the basic functionalities and components of the starter kit, guiding users through initial interactions.

## 5. PROGRAMMING LANGUAGES SUPPORTED

---

The LAFVIN Basic Starter Kit supports multiple programming languages, making it versatile for various learning levels:

- **MicroPython:** A lean and efficient implementation of the Python 3 programming language, optimized for microcontrollers.
- **C/C++:** A powerful and widely used programming language for embedded systems, offering high performance and control.
- **Piper Make:** A visual block-based programming environment, ideal for beginners to learn programming concepts without complex syntax.

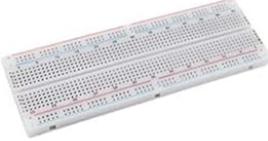
			
Raspberry Pi Pico x1	IIC LCD 1602 x1	830 Tie-Points Breadboard x1	Obstacle Avoidance Module x1
			
HC-SR501 PIR Motion Sensor x1	SG90 Servo x1	Potentiometer (10k) x1	Passive Buzzer x1 Active Buzzer x1
			
Resistor-220R/1k/10k x30	Micro-USB Cable x1	Button Switch x8	WS2812 RGB 8 LEDs Strip x1
			
F-M DuPont Cable x10	65 Jumper Wire x1	Thermistor x1 Tilt Switch x1	Photoresistor x2 S8050 Transistor x2
			
LED-Red x5	LED-Yellow x5	LED-Green x5	LED-White x1

Figure 5.1: Supported Programming Languages.

This image highlights the three primary programming languages—MicroPython, Piper Make, and C/C++—that can be used with the Raspberry Pi Pico, catering to different skill levels.

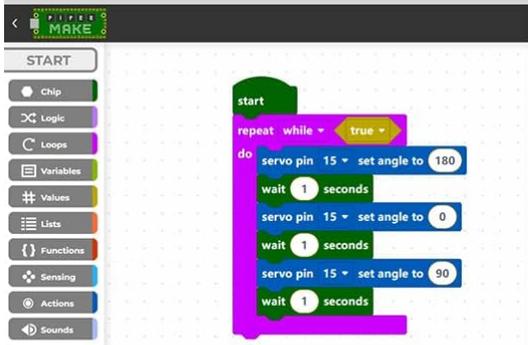
# Support 3 Popular Programming Languages

```
File Edit View Run Tools Help
[untitled] [code.py] - hello_led.py - use potentiometer to control Servo.py
7  servoPin.freq(50)
8
9
10 def servo(degrees):
11     # limit degrees between 0 and 180
12     if degrees > 180: degrees=180
13     if degrees < 0: degrees=0
14     # set max and min duty
15     maxDuty=9000
16     minDuty=1000
17     # use potentiometer to control Servo potentiometer to control Servo
18     # new duty is between min and max duty in proportion to its value
19     newDuty=minDuty+(maxDuty-minDuty)*(degrees/180)
20     # servo PWM value is set
21     servoPin.duty_u16(int(newDuty))
22
23 while True:
24     value=adc.read_u16()
25     print(value)
26     degree=value*180/65500
```



## MicroPython

MicroPython is a lean and efficient implementation of the Python 3 programming language.



## Piper Make

Piper Make is a super easy and fun way to make projects using Raspberry Pi Pico.

```
IRemote(int pin):
ErrorStatus decode():
void begin():
void end():
void loop():
boolean keyPressed(unsigned char r):
// void resume():

int8_t decode_type; // NEC, SONY, RC5, UNKNOWN
unsigned long value; // Decoded value
uint8_t bits; // Number of bits in decoded value
volatile uint8_t *rawBuf; // Raw intervals in .5 us ticks
int rawLen; // Number of records in rawBuf.
String getString():
unsigned char getCode():
String getKeyMap(byte keycode, byte ir_type = 1):
byte getIrKey(byte keycode, byte ir_type = 1):
void sendString(String s):
void sendString(float v):
void sendNEC(unsigned long data, int nbits):
void sendRaw(unsigned int buf[], int len, uint8_t Hz):
void enableIROut(uint8_t kHz):
void enableIRIn():
```



## C/ C++

C++ is a human-readable programming language. When you create a code, it is processed and compiled to machine language.

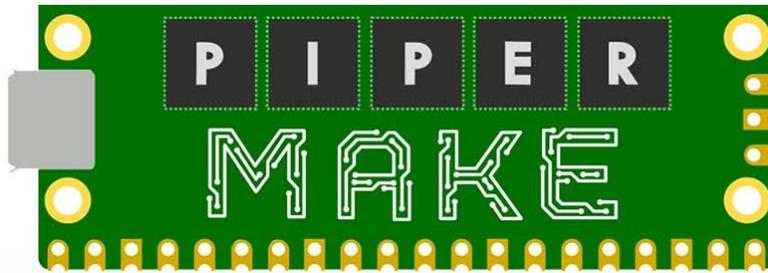
Figure 5.2: Piper Make Interface.

This image shows the user-friendly graphical interface of Piper Make, demonstrating its block-based approach to programming, which simplifies coding for beginners.

## 6. EXAMPLE PROJECTS

The kit provides resources to build various projects, including:

- LEDs Strip Control
- Potentiometer Control Servo
- Button Controlled LED
- Photoresistor Controlled LED
- Music Buzzer
- Motion Detection using PIR Sensor
- Display CPU Temperature using LCD1602



**Piper Make** allows children to learn and absorb the basics of programming in the electronics industry as they learn how to build their own computers.

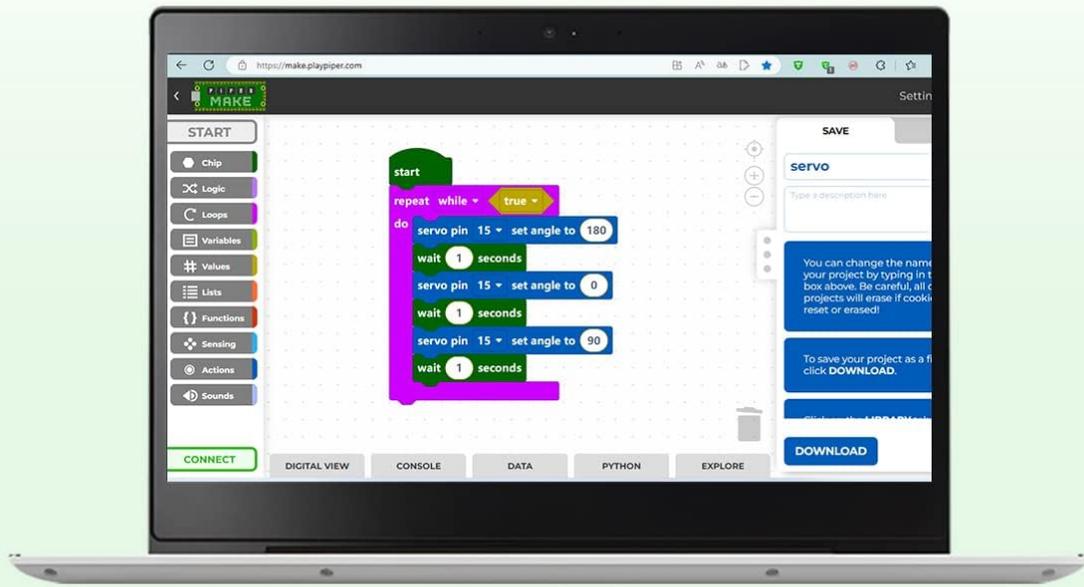


Figure 6.1: Various Project Examples.

This image showcases four different projects that can be created with the kit, demonstrating the versatility of the components for various electronic applications.

Your browser does not support the video tag.

Video 6.1: GeeekPi Raspberry Pi Pico Basic Starter Kit Tutorial.

This extensive tutorial video from GeeekPi demonstrates various projects and detailed setup procedures for a Raspberry Pi Pico basic starter kit, offering valuable insights for users.

## 7. SPECIFICATIONS

Feature	Detail
Processor	Cortex (RP2040)
RAM	LPDDR2
Operating System	MicroPython
Item Weight	11.3 ounces

Product Dimensions	5.51 x 3.35 x 1.42 inches
Number of Processors	1

## 8. TROUBLESHOOTING

---

If you encounter issues while using the kit, consider the following common troubleshooting steps:

- **Wiring Check:** Double-check all connections on the breadboard against the schematic in the tutorial. Loose or incorrect wires are a common cause of problems.
- **Code Verification:** Ensure your code is correctly written and uploaded. Pay attention to syntax, indentation, and pin assignments.
- **Power Supply:** Confirm the Raspberry Pi Pico is receiving adequate power via the Micro-USB cable.
- **Component Functionality:** Test individual components if possible to rule out faulty parts.
- **Software Environment:** Verify that your IDE (e.g., Thonny) and MicroPython firmware are correctly installed and configured.
- **Online Resources:** Refer to the official Raspberry Pi Pico documentation and online forums for further assistance.

## 9. WARRANTY AND SUPPORT

---

For warranty information, technical support, or further inquiries, please refer to the contact information provided by LAFVIN at the time of purchase or visit the official LAFVIN website. Keep your purchase receipt for warranty claims.