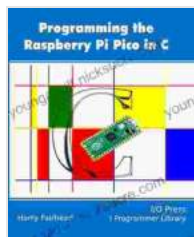


Programming Raspberry Pi Pico in C++: A Comprehensive Guide

The Raspberry Pi Pico is a tiny and powerful microcontroller that is perfect for beginners and experienced makers alike. It is based on the RP2040 chip, which features dual-core ARM Cortex-M0+ processors, 256KB of RAM, and 2MB of flash memory. The Pico also has a variety of built-in peripherals, including GPIO pins, I2C, SPI, and UART.



Programming The Raspberry Pi Pico In C by Harry Fairhead

★★★★☆ 4.1 out of 5

Language : English
File size : 30387 KB
Screen Reader : Supported
Print length : 373 pages
Lending : Enabled
X-Ray for textbooks : Enabled



In this guide, we will learn how to program the Raspberry Pi Pico in C++. We will cover the basics of C++ programming, as well as how to use the Pico's peripherals. By the end of this guide, you will be able to write your own C++ programs for the Raspberry Pi Pico.

Prerequisites

Before you begin, you will need the following:

- A Raspberry Pi Pico

- A USB cable
- A computer with a C++ compiler

Setting Up Your Development Environment

The first step is to set up your development environment. We will be using the Arduino IDE, which is a free and open-source software that is specifically designed for programming microcontrollers.

To install the Arduino IDE, follow these steps:

1. Go to the Arduino website and download the latest version of the IDE. 2. Install the Arduino IDE on your computer. 3. Open the Arduino IDE and go to **File > Preferences**. 4. In the **Additional Boards Manager URLs** field, add the following URL:

```
https://github.com/raspberrypi/pico/raw/master/package\_rp2040\_
```

Click **OK**. 6. Go to **Tools > Board > Boards Manager**. 7. In the search bar, type "Raspberry Pi Pico". 8. Select the **Raspberry Pi Pico** board and click **Install**.

Writing Your First C++ Program

Now that you have your development environment set up, let's write our first C++ program for the Raspberry Pi Pico.

Open the Arduino IDE and create a new sketch. Copy and paste the following code into the sketch:

```
cpp #include
```

```
void setup(){pinMode(LED_BUILTIN, OUTPUT); }
```

```
void loop(){digitalWrite(LED_BUILTIN, HIGH); delay(500);  
digitalWrite(LED_BUILTIN, LOW); delay(500); }
```

This program will blink the LED on the Raspberry Pi Pico on and off every second.

To upload the program to the Pico, click the **Upload** button in the Arduino IDE. Once the program has been uploaded, the LED on the Pico should start blinking.

Using the Pico's Peripherals

The Raspberry Pi Pico has a variety of built-in peripherals that can be used to interact with the outside world. In this section, we will learn how to use some of the most common peripherals, including GPIO pins, I2C, SPI, and UART.

GPIO Pins

GPIO pins are general-purpose input/output pins that can be used to connect the Pico to other devices. The Pico has 26 GPIO pins, which are located on the two sides of the board.

To use a GPIO pin, you first need to set it up as an input or output. You can do this using the `pinMode()` function.

```
cpp pinMode(GPIO_PIN, INPUT); pinMode(GPIO_PIN, OUTPUT);
```

Once you have set up a GPIO pin, you can read or write to it using the `digitalRead()` and `digitalWrite()` functions.

```
cpp int value = digitalRead(GPIO_PIN); digitalWrite(GPIO_PIN, HIGH);  
digitalWrite(GPIO_PIN, LOW);
```

I2C

I2C is a serial communication protocol that is used to connect devices to each other. The Pico has two I2C buses, which are located on the top and bottom of the board.

To use I2C, you first need to create an I2C object. You can do this using the **Wire** library.

```
cpp #include
```

```
Wire.begin();
```

Once you have created an I2C object, you can use it to send and receive data from other devices. You can do this using the **write()** and **read()** functions.

```
cpp Wire.write(data); Wire.read(data);
```

SPI

SPI is a serial communication protocol that is used to connect devices to each other at high speeds. The Pico has one SPI bus, which is located on the bottom of the board.

To use SPI, you first need to create an SPI object. You can do this using the **SPI** library.

```
cpp #include
```

```
SPI.begin();
```

Once you have created an SPI object, you can use it to send and receive data from other devices. You can do this using the `transfer()` function.

```
cpp SPI.transfer(data);
```

UART

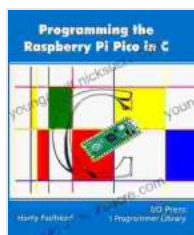
UART is a serial communication protocol that is used to connect devices to each other over long distances. The Pico has one UART bus, which is located on the bottom of the board.

To use UART, you first need to create a UART object. You can do this using the `Serial` library.

```
cpp #include
```

```
Serial.begin();
```

Once you have created a UART object, you can use it to send and receive data



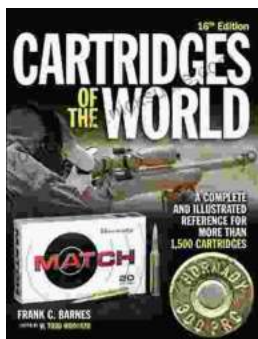
Programming The Raspberry Pi Pico In C by Harry Fairhead

★★★★☆ 4.1 out of 5

Language : English
File size : 30387 KB
Screen Reader : Supported
Print length : 373 pages
Lending : Enabled
X-Ray for textbooks : Enabled

FREE

DOWNLOAD E-BOOK



Delve into the Comprehensive World of Cartridges: A Comprehensive Review of Cartridges of the World 16th Edition

In the realm of firearms, cartridges stand as the linchpins of operation, propelling projectiles towards their targets with precision and power. Cartridges of the World, a...



Tales From The San Francisco 49ers Sideline: A Look Inside The Team's Inner Sanctum

The San Francisco 49ers are one of the most iconic franchises in the NFL. With five Super Bowl victories, the team has a rich history and tradition that is unmatched by many...