CEC Continuing Education Center

DesignNews

Getting Started with the Raspberry Pi Pico

DAY 3 : Interfacing with Raspberry Pi Pico Peripherals

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

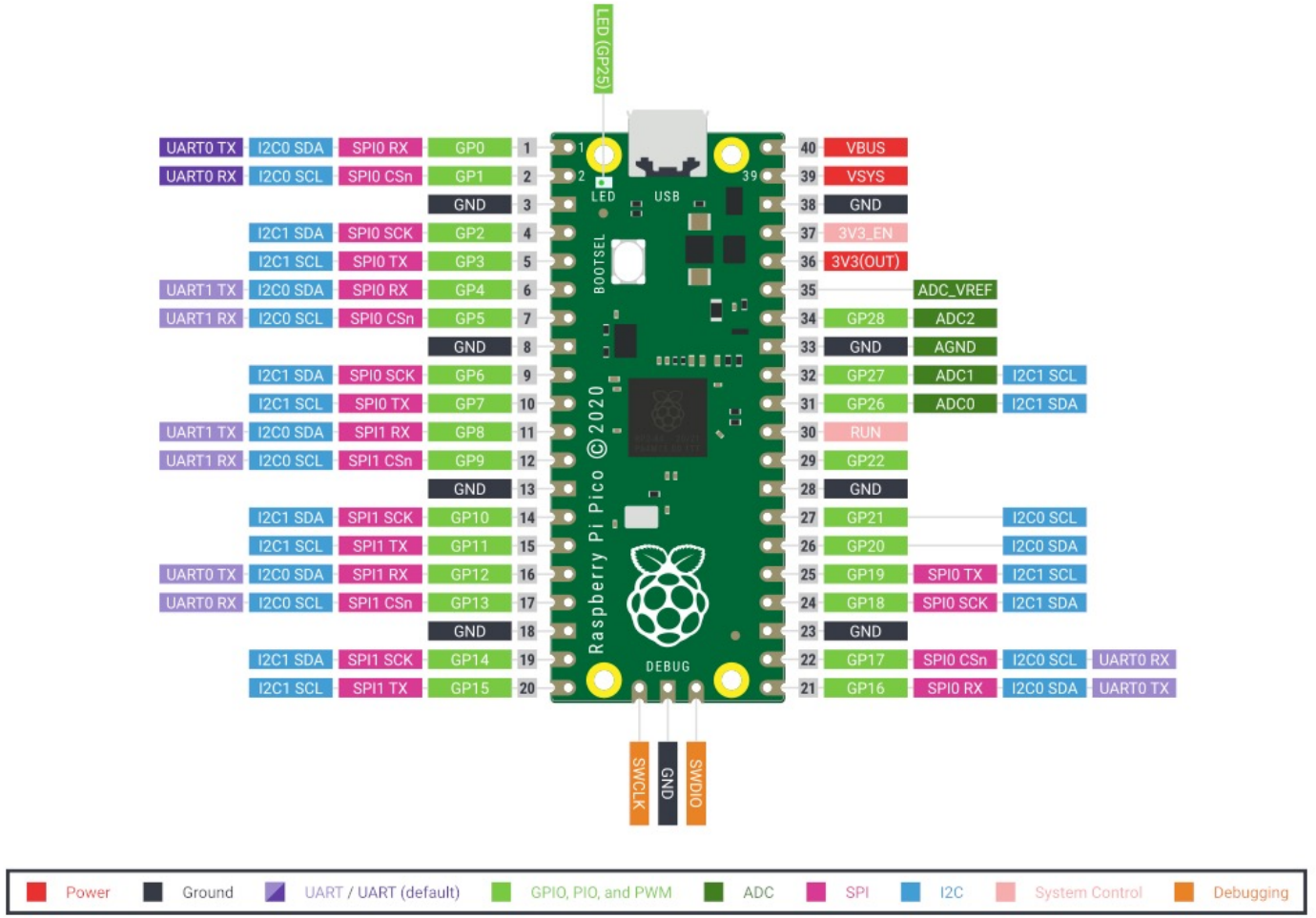
Course Sessions

- Introduction to the Raspberry Pi Pico
- Writing your First Raspberry Pi Pico Application
- Interfacing with Raspberry Pi Pico Peripherals
- Designing Multicore Raspberry Pi Pico Applications
- Using MicroPython on the Raspberry Pi Pico

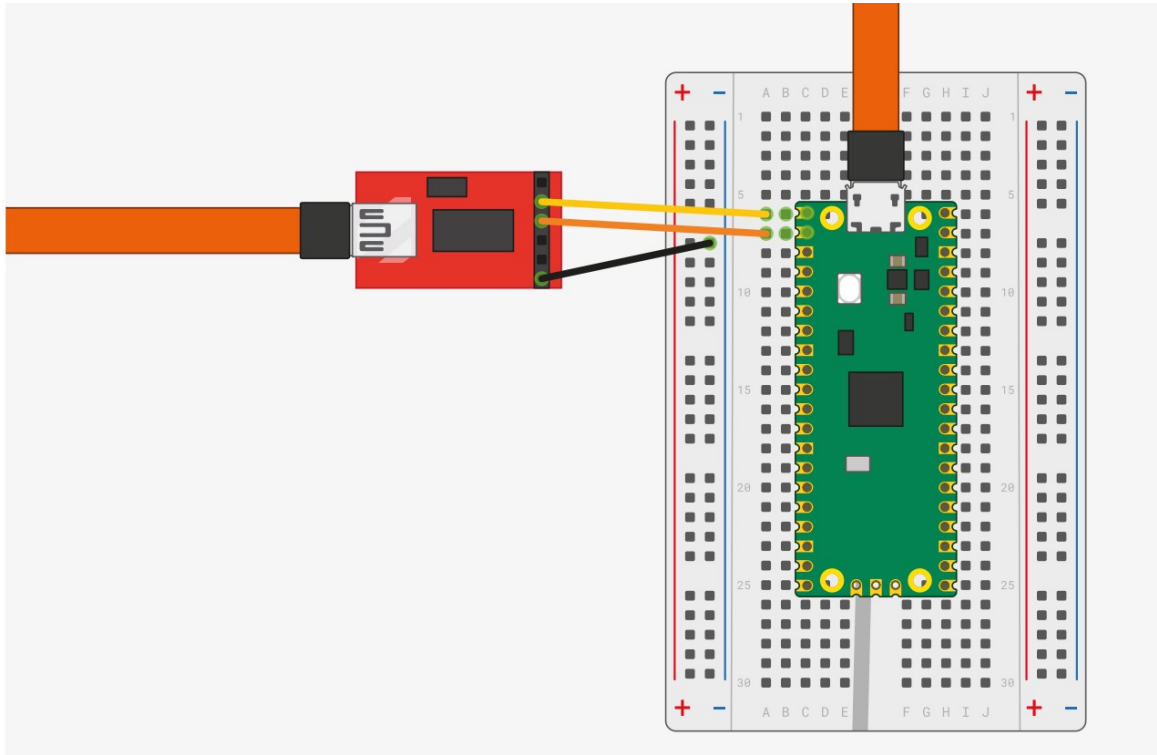
1 The Raspberry Pi Pico Peripherals

The Module Overview

- GPIO (28)
  - Any GPIO can be PWM
- UART (2)
- I2C (2)
- SPI(2)
- Analog (3)



Hello World Connection Diagram



Hello World!

Hello World has 2 options:

- USB
- UART

Common Code, Makefile Controlled:

```
# enable usb output, disable uart output
pico_enable_stdio_usb(hello_usb 1)
pico_enable_stdio_uart(hello_usb 0)
```

```
#include <stdio.h>
#include "pico/stdlib.h"

int main() {
    stdio_init_all();
    while (true) {
        printf("Hello, world!\n");
        sleep_ms(1000);
    }
    return 0;
}
```

Which interface do you prefer for ”Hello World”?
- USB
- UART

In the chat, explain why

2 Analog to Digital Conversion

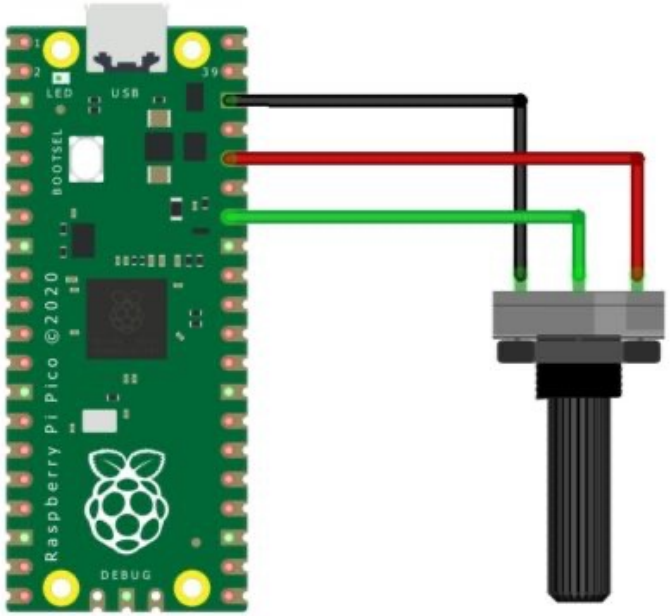
ADC Physical Characteristics

- ADC0 – GP26
- ADC1 – GP27
- ADC2 – GP28

Max Voltage Reference 3.3V

12-bit Resolution

ADC Connection Diagram



ADC Example Source Code

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"
#include "hardware/adc.h"

int main()
{
    stdio_init_all();
    printf("ADC Example, measuring GPIO26\n");

    adc_init();

    adc_gpio_init(28);
...
```

```
    // Select ADC input 0 (GPIO28)
    adc_select_input(0);

    while (1)
    {
        // 12-bit conversion
        const float conversion_factor = 3.3f / (1 << 12);
        uint16_t result = adc_read();

        printf("Raw value: 0x%03x, voltage: %f V\n", result,
                result * conversion_factor);

        sleep_ms(500);
    }
}
```

3 Pulse Width Modulation (PWM)

PWM Physical Characteristics

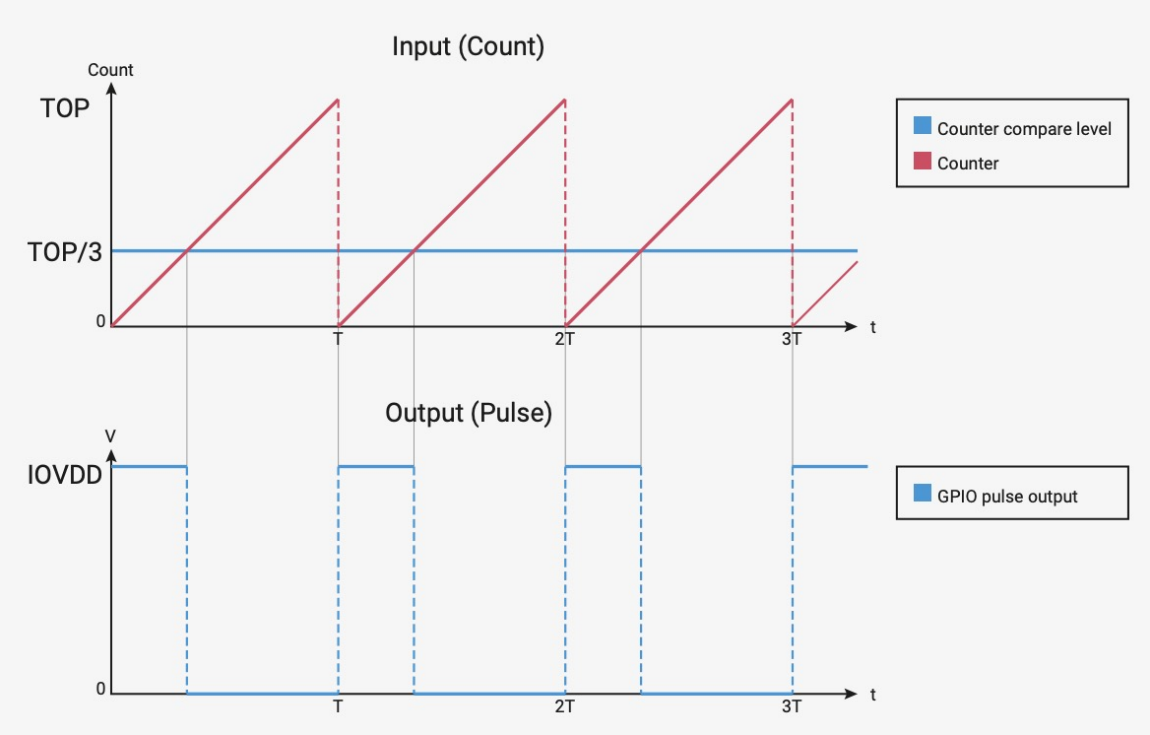
- Any GPIO pin can generate PWM signals
- 0 – 3.3 Volts

All 30 GPIO pins on RP2040 can be used for PWM:

GPIO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PWM Channel	0A	0B	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A	7B
GPIO	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
PWM Channel	0A	0B	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B		

https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf
Section 4.5.2

ADC Example Source Code



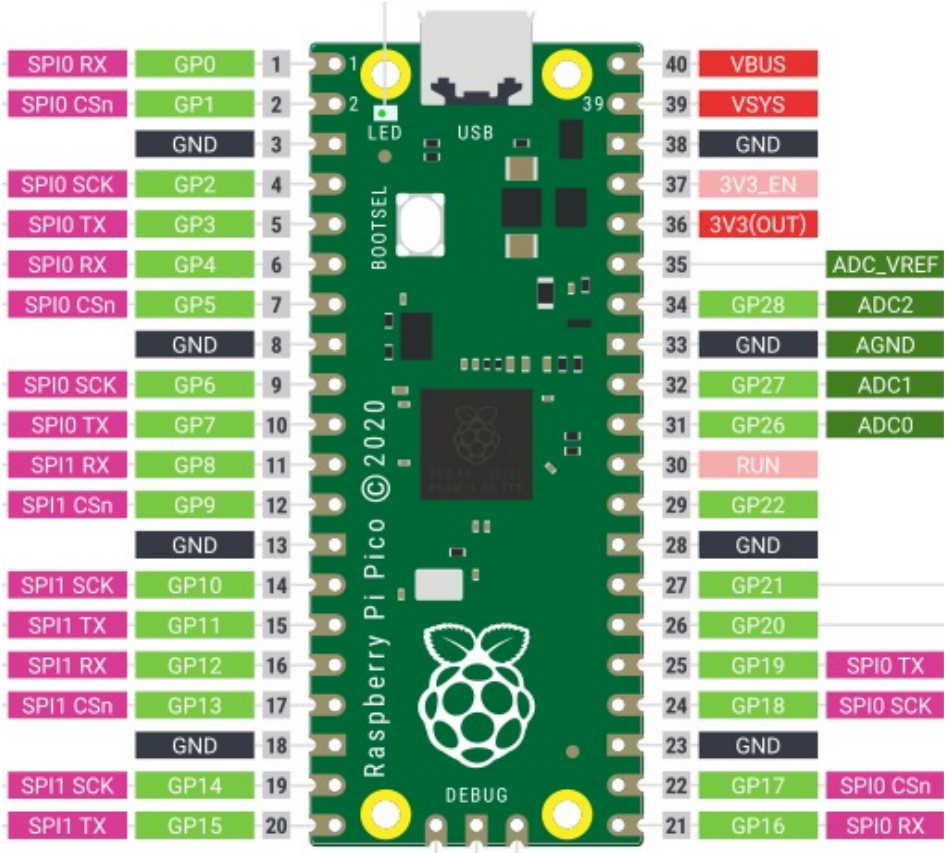

```
#include "pico/stdlib.h"
#include "hardware/pwm.h"

int main()
{
    // Tell GPIO 0 they are allocated to the PWM
    gpio_set_function(0, GPIO_FUNC_PWM);
    gpio_set_function(1, GPIO_FUNC_PWM);

    // Find out which PWM slice is connected to GPIO 0 (it's slice 0)
    uint slice_num = pwm_gpio_to_slice_num(0);

    // Set period of 4 cycles (0 to 3 inclusive)
    pwm_set_wrap(slice_num, 3);

    // Set channel A output high for one cycle before dropping
    pwm_set_chan_level(slice_num, PWM_CHAN_A, 1);

    // Set initial B output high for three cycles before dropping
    pwm_set_chan_level(slice_num, PWM_CHAN_B, 3);

    // Set the PWM running
    pwm_set_enabled(slice_num, true);
}
```

4 Serial Peripheral Interface (SPI)

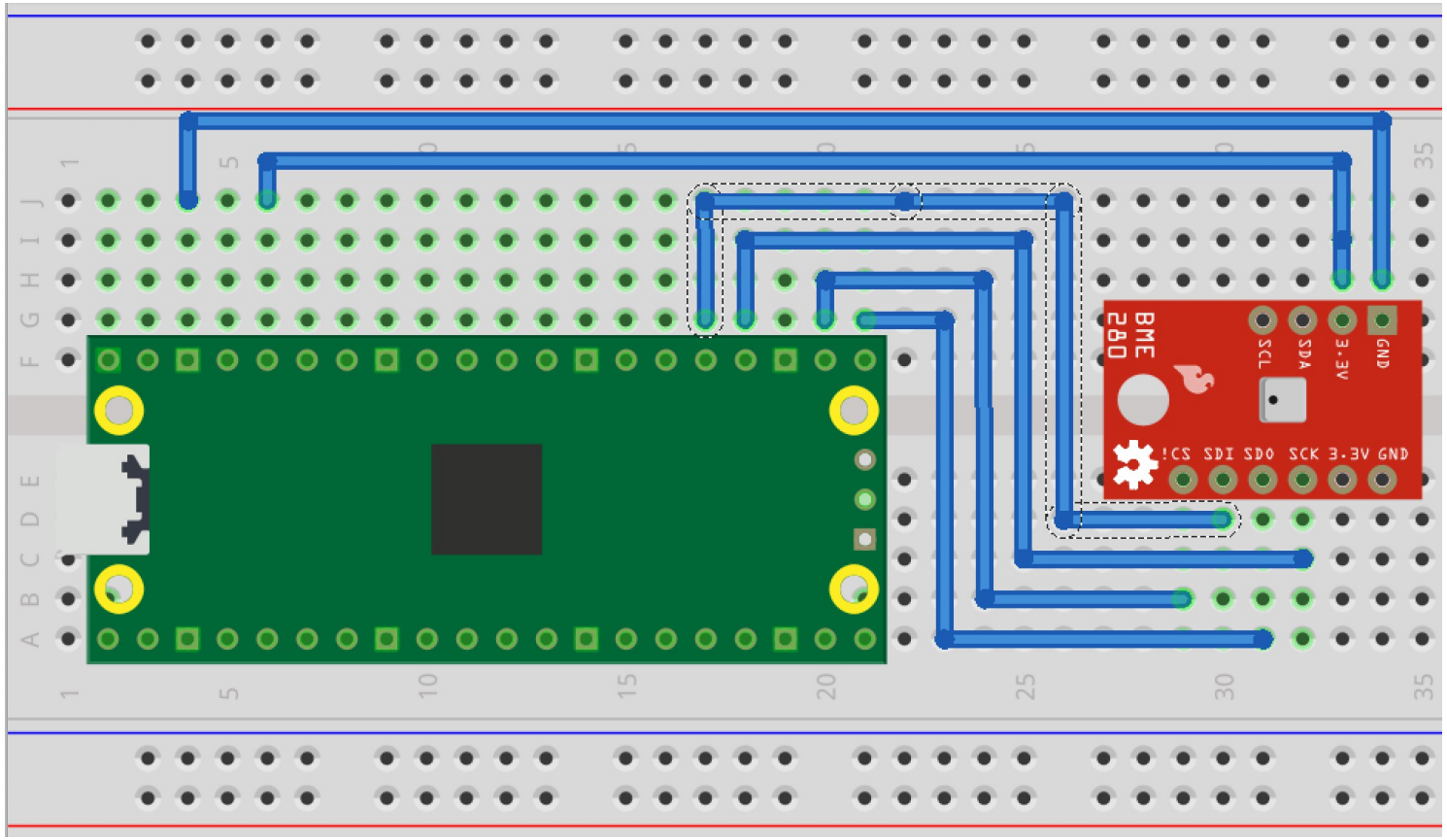
SPI Physical Characteristics

- SPI0 RX – GP0, GP4, GP16
- SPI0 TX – GP3, GP7, GP19
- SPI0 SCK – GP2, GP6, GP18

- SPI1 RX – GP8, GP12
- SPI1 TX – GP11, GP15
- SPI1 SCK – GP10, GP14



SPI Connection Diagram to a BME280



SPI Example Source Code

```
#include <stdio.h>
#include <string.h>
#include "pico/stdlib.h"
#include "pico/binary_info.h"
#include "hardware/spi.h”

int main() {
    stdio_init_all();

    // This example will use SPI0 at 0.5MHz.
    spi_init(spi_default, 500 * 1000);

    gpio_set_function(PICO_DEFAULT_SPI_RX_PIN, GPIO_FUNC_SPI);
    gpio_set_function(PICO_DEFAULT_SPI_SCK_PIN, GPIO_FUNC_SPI);
    gpio_set_function(PICO_DEFAULT_SPI_TX_PIN, GPIO_FUNC_SPI);

    bi_decl(bi_3pins_with_func(PICO_DEFAULT_SPI_RX_PIN,
            PICO_DEFAULT_SPI_TX_PIN, PICO_DEFAULT_SPI_SCK_PIN,
            GPIO_FUNC_SPI));
```

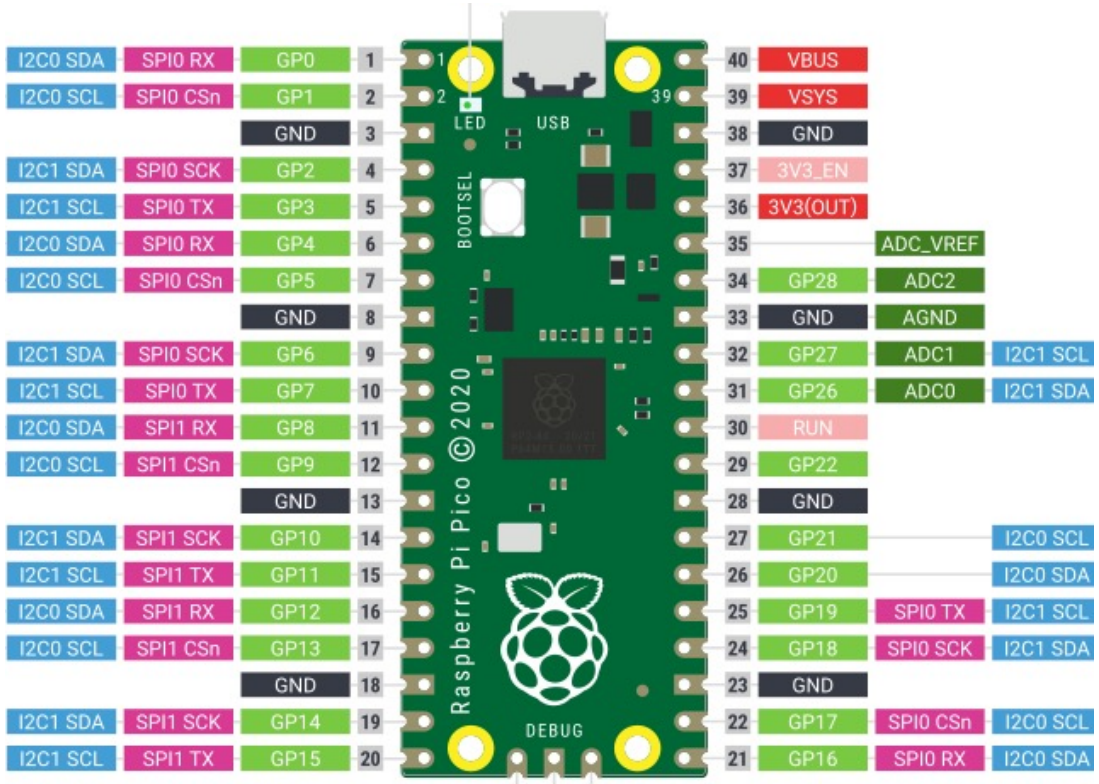
```
    gpio_init(PICO_DEFAULT_SPI_CSN_PIN);
    gpio_set_dir(PICO_DEFAULT_SPI_CSN_PIN, GPIO_OUT);
    gpio_put(PICO_DEFAULT_SPI_CSN_PIN, 1);

    // Make the CS pin available to picotool
    bi_decl(bi_1pin_with_name(PICO_DEFAULT_SPI_CSN_PIN, "SPI CS"));

    while (1) {
        bme280_read_raw(&humidity, &pressure, &temperature);

        pressure = compensate_pressure(pressure);
        temperature = compensate_temp(temperature);
        humidity = compensate_humidity(humidity);

        printf("Humidity = %.2f%%\n", humidity / 1024.0);
        printf("Pressure = %dPa\n", pressure);
        printf("Temp. = %.2fC\n", temperature / 100.0);

        sleep_ms(1000);
    }
```

5 Inter-IC (I2C) Communication

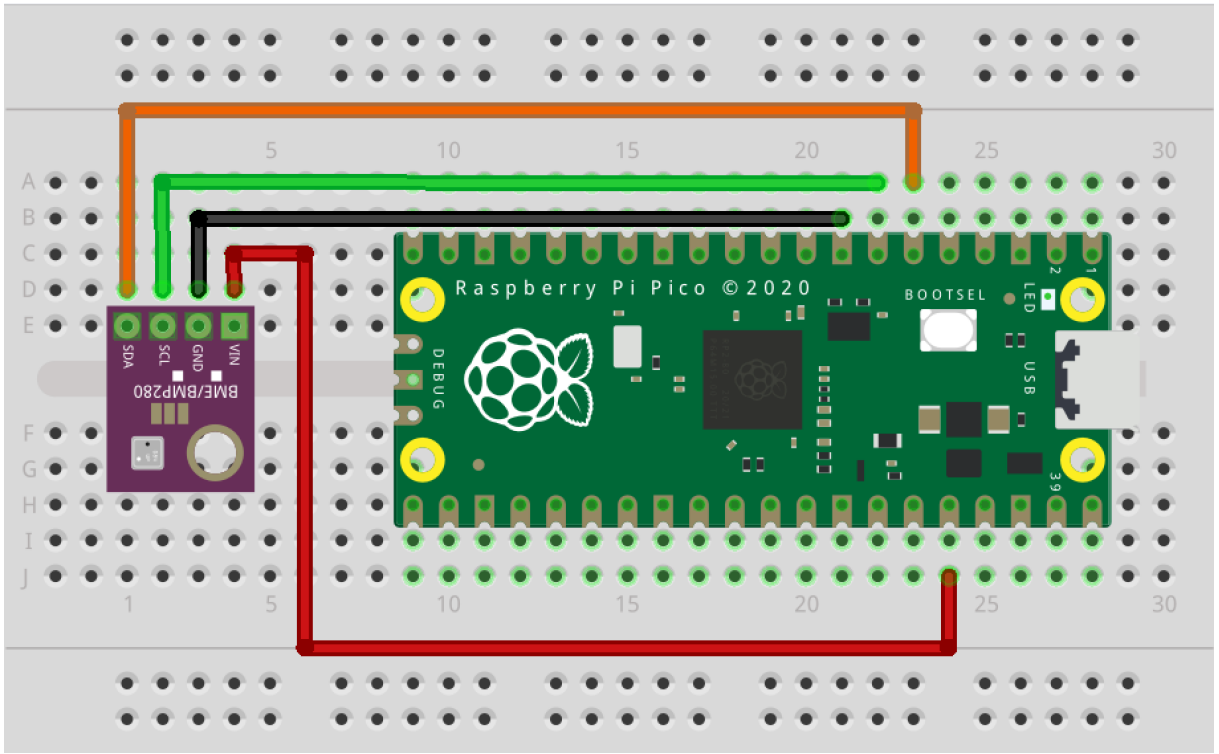
ADC Physical Characteristics

- I2C0 SDA – GP0, GP4, GP8, GP12, GP16, GP20
- I2C0 SCL – GP1, GP5, GP9, GP13, GP17, GP21

- I2C1 SDA – GP2, GP6, GP10, GP14, GP18, GP26
- I2C1 SCL – GP3, GP7, GP11, GP15, GP19, GP27



I2C Connection Diagram BME280



I2C Example Source Code

```
#include <stdio.h>

#include "hardware/i2c.h"
#include "pico/binary_info.h"
#include "pico/stdlib.h"

int main()
{
    stdio_init_all();

    i2c_init(i2c_default, 100 * 1000);

    gpio_set_function(PICO_DEFAULT_I2C_SDA_PIN, GPIO_FUNC_I2C);
    gpio_set_function(PICO_DEFAULT_I2C_SCL_PIN, GPIO_FUNC_I2C);
    gpio_pull_up(PICO_DEFAULT_I2C_SDA_PIN);
    gpio_pull_up(PICO_DEFAULT_I2C_SCL_PIN);

    // configure BMP280
    bmp280_init();
```

```
while (1) {
        bmp280_read_raw(&raw_temperature, &raw_pressure);
        int32_t temperature =
                bmp280_convert_temp(raw_temperature,
                                    &params);

        int32_t pressure =
                 bmp280_convert_pressure(raw_pressure,
                 raw_temperature, &params);

        printf("Pressure = %.3f kPa\n", pressure / 1000.f);
        printf("Temp. = %.2f C\n", temperature / 100.f);
        // poll every 500ms
        sleep_ms(500);
}
```

6 Going Further

Thank you for attending

Please consider the resources below:

- www.beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Getting Started with the Raspberry Pi Pico

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets