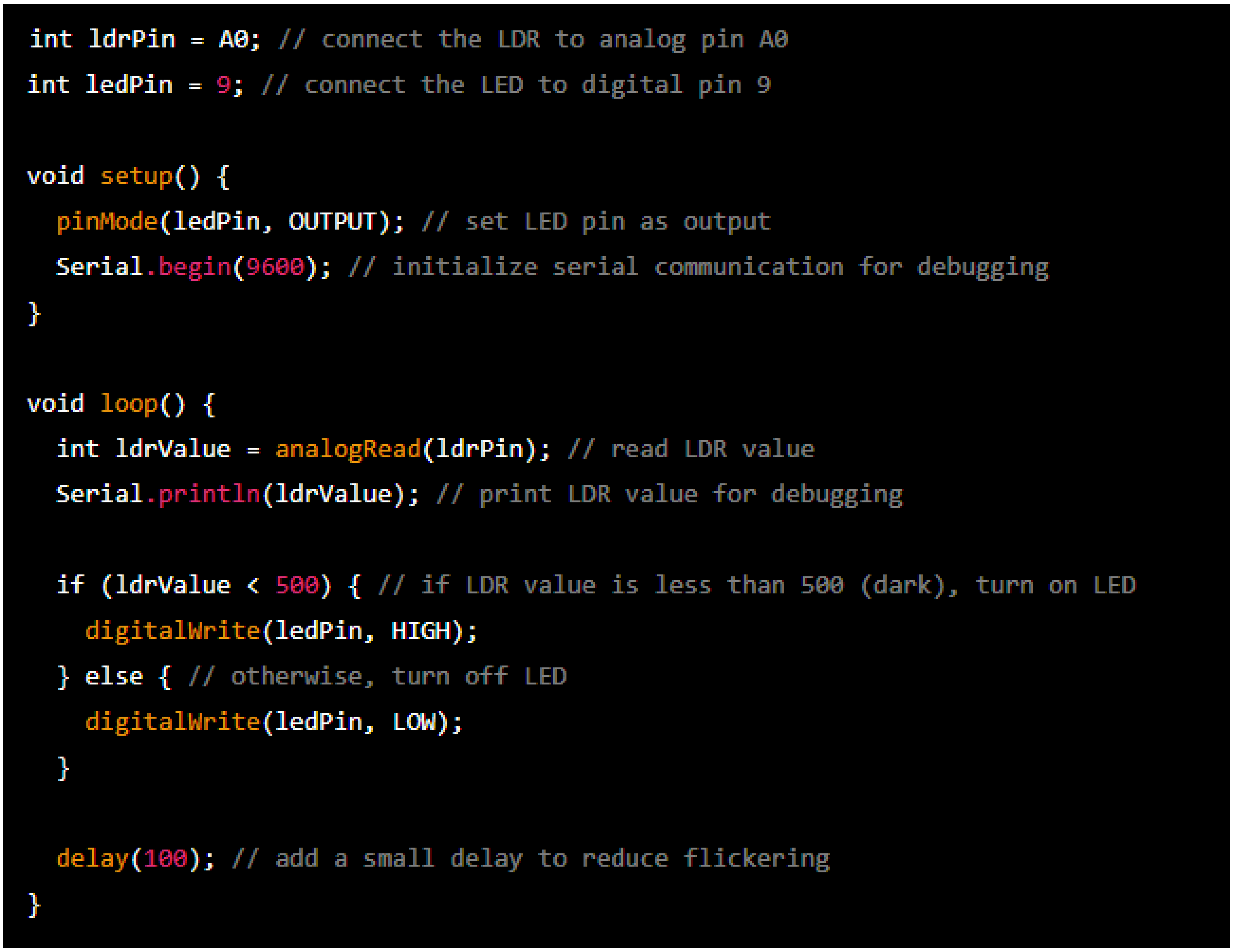
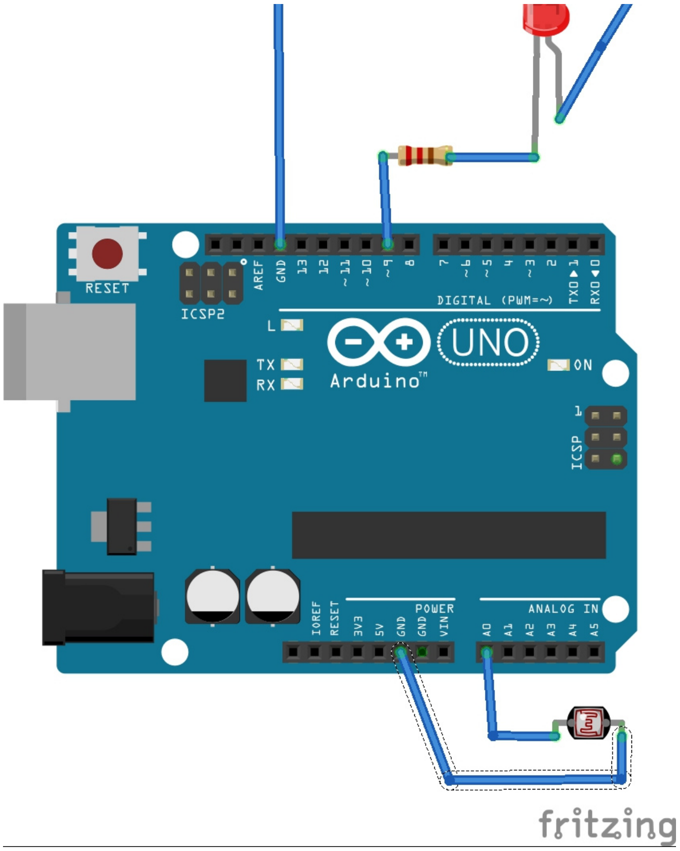
# Embedded Systems

Write a code in Arduino IDE to interface a LDR and a LED which toggles according to (for Arduino UNO Raspberry Pi pico / Espressif Systems

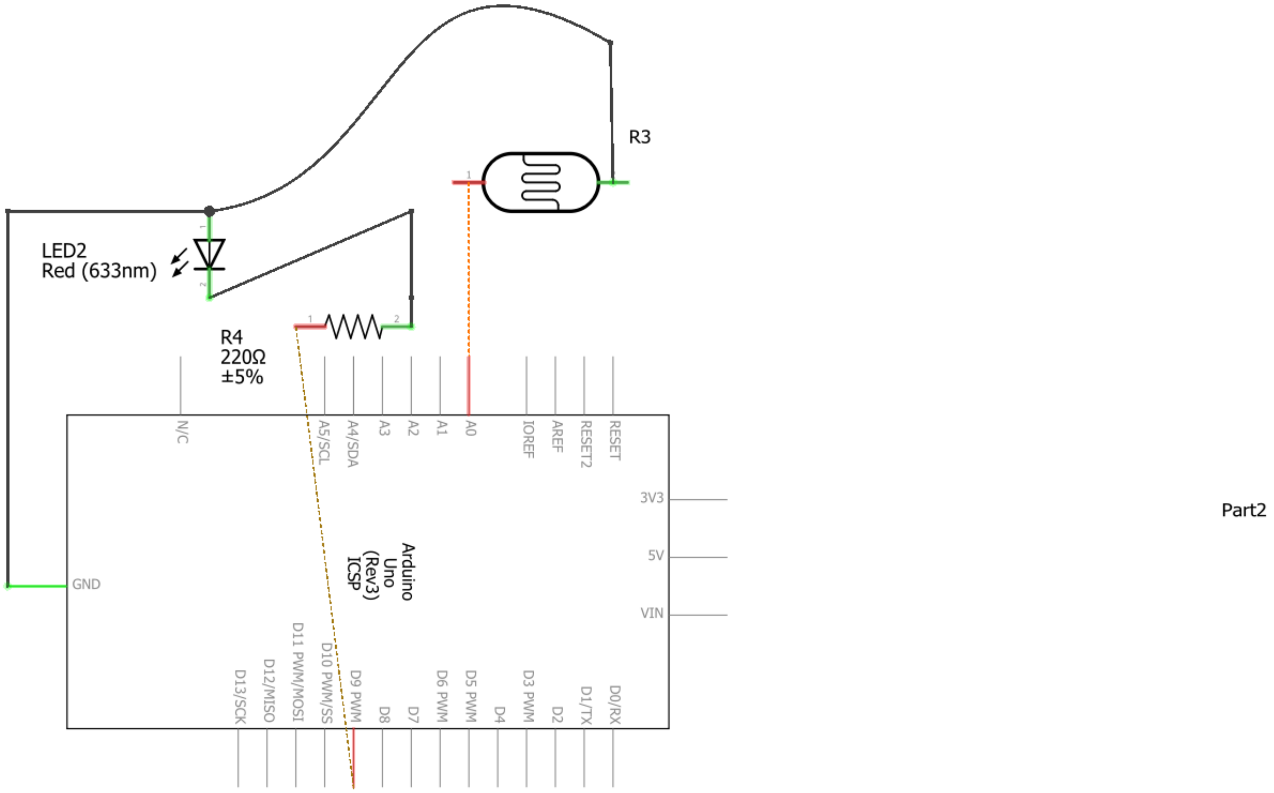


```
int ldrPin = A0; // connect the LDR to analog pin A0
int ledPin = 9; // connect the LED to digital pin 9

void setup() {
  pinMode(ledPin, OUTPUT); // set LED pin as output
  Serial.begin(9600); // initialize serial communication for debugging
}

void loop() {
  int ldrValue = analogRead(ldrPin); // read LDR value
  Serial.println(ldrValue); // print LDR value for debugging

  if (ldrValue < 500) { // if LDR value is less than 500 (dark), turn on LED
    digitalWrite(ledPin, HIGH);
  } else { // otherwise, turn off LED
    digitalWrite(ledPin, LOW);
  }

  delay(100); // add a small delay to reduce flickering
}
```

## Circuit Diagram



SCHEMATIC DIAGRAM



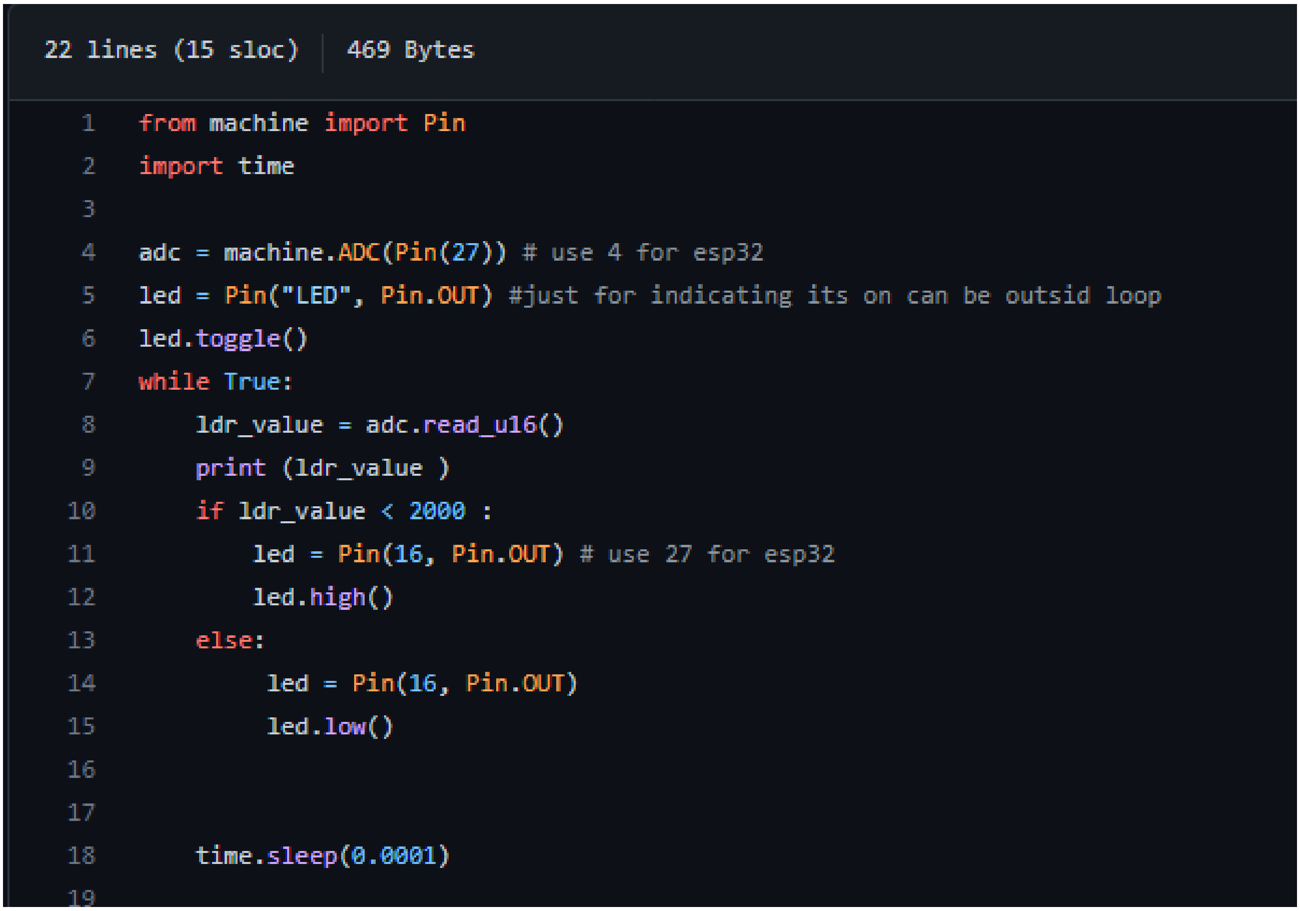
fritzing

# Interfacing ESP8266/12E/ESP32

# or Raspberry Pi Pico with LDR and LED code in Micropython

Identify the overarching sentiment you want to communicate to your target audience that ties this project's different copy requirements together.

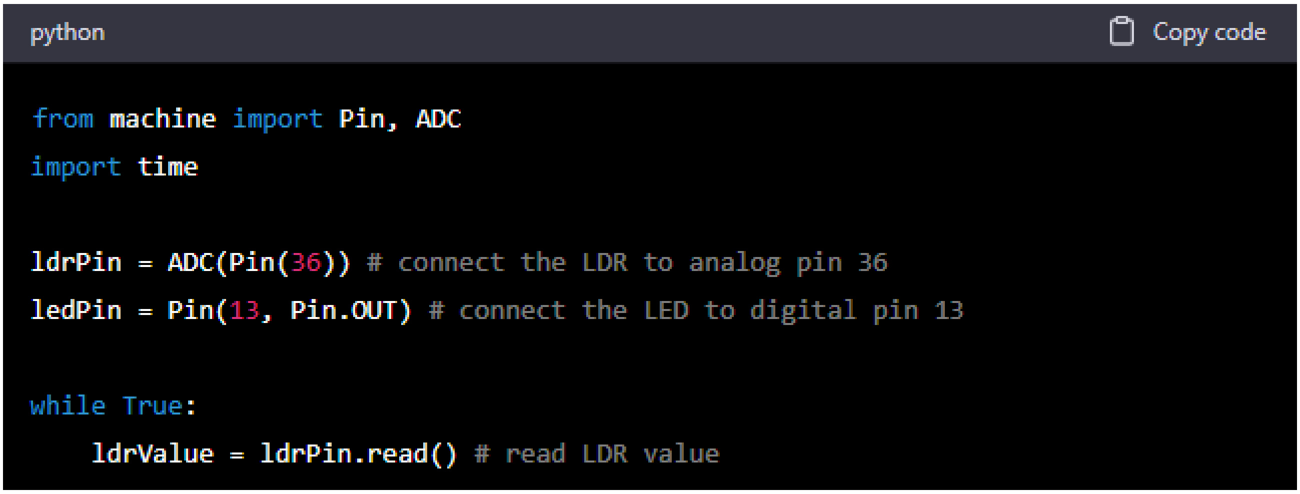
RASPBERRY PI PICO CODE : -



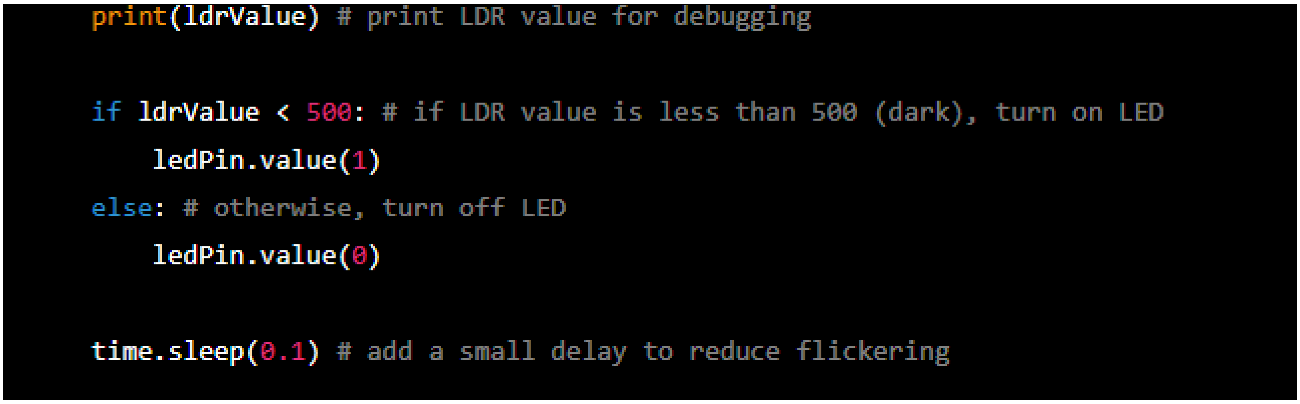
22 lines (15 sloc) | 469 Bytes

```python
from machine import Pin
import time

adc = machine.ADC(Pin(27)) # use 4 for esp32
led = Pin("LED", Pin.OUT) #just for indicating its on can be outsid loop
led.toggle()
while True:
    ldr_value = adc.read_u16()
    print (ldr_value )
    if ldr_value < 2000 :
        led = Pin(16, Pin.OUT) # use 27 for esp32
        led.high()
    else:
         led = Pin(16, Pin.OUT)
         led.low()


    time.sleep(0.0001)
```

ESP32 code : -

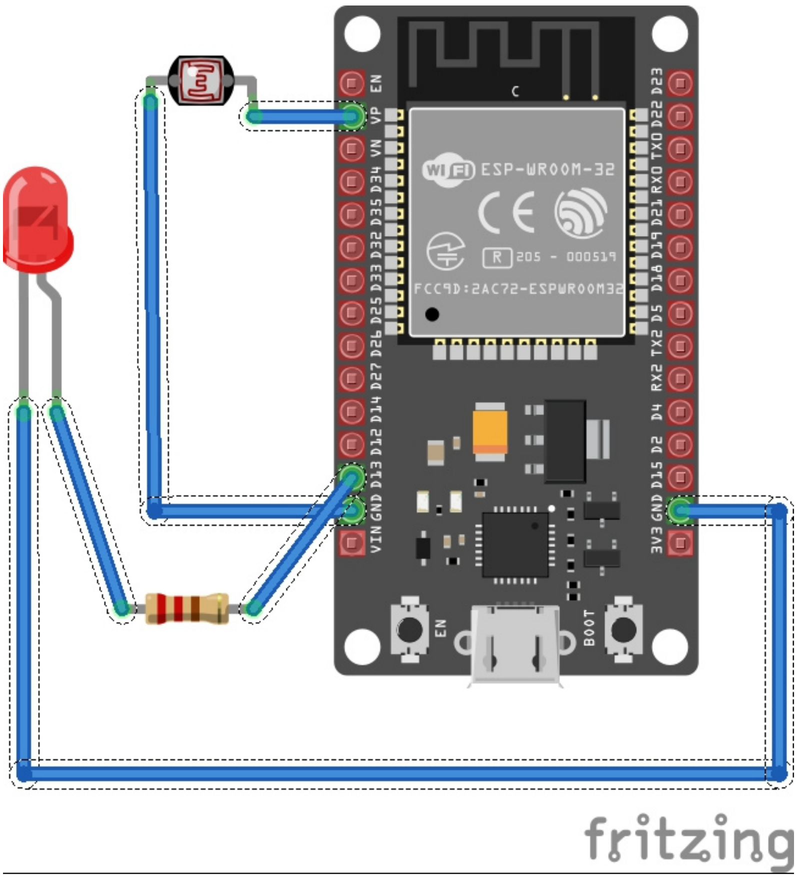


```python
from machine import Pin, ADC
import time

ldrPin = ADC(Pin(36)) # connect the LDR to analog pin 36
ledPin = Pin(13, Pin.OUT) # connect the LED to digital pin 13

while True:
    ldrValue = ldrPin.read() # read LDR value
```

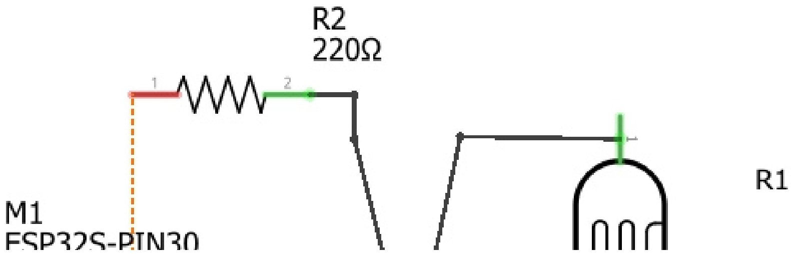


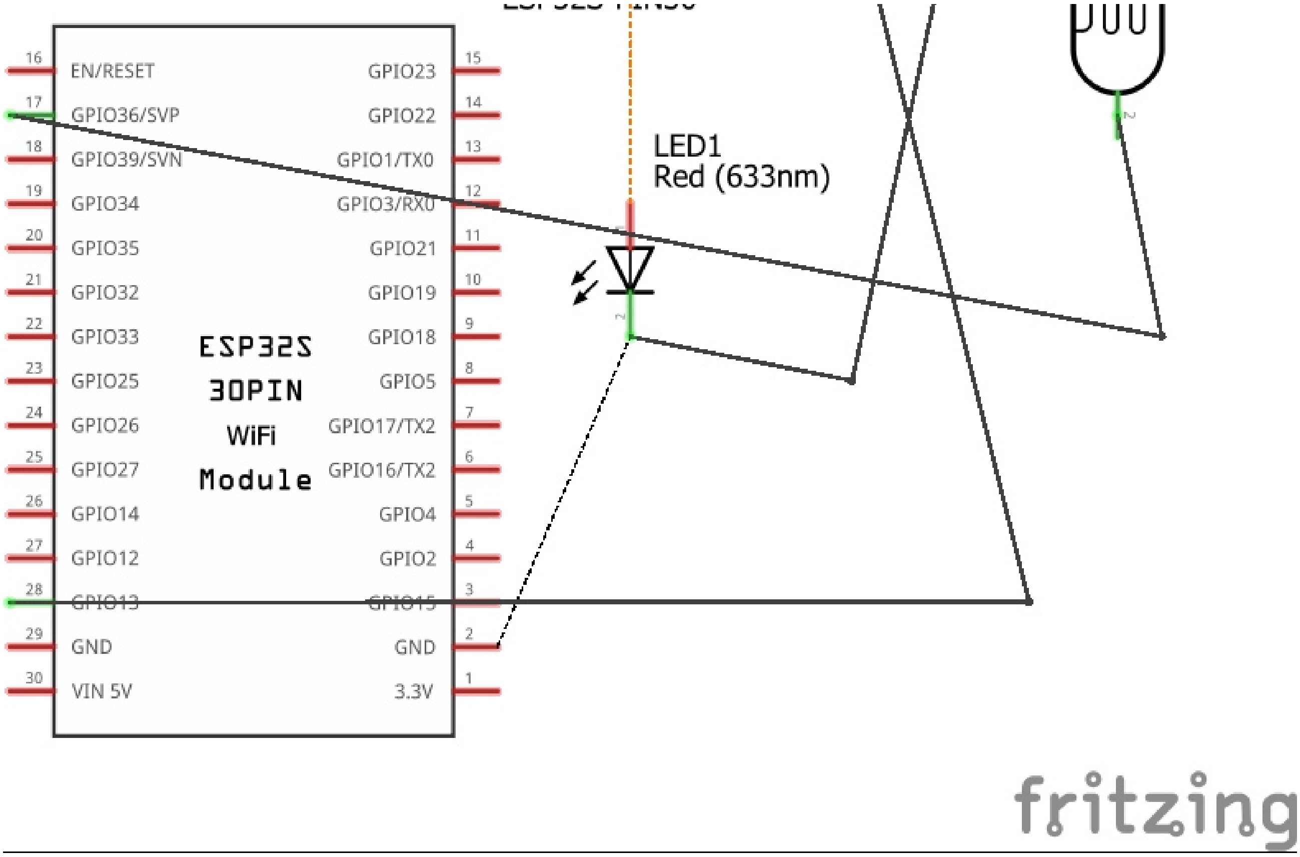
```
    print(ldrValue) # print LDR value for debugging

    if ldrValue < 500: # if LDR value is less than 500 (dark), turn on LED
        ledPin.value(1)
    else: # otherwise, turn off LED
        ledPin.value(0)

    time.sleep(0.1) # add a small delay to reduce flickering
```

## Circuit Diagram :-

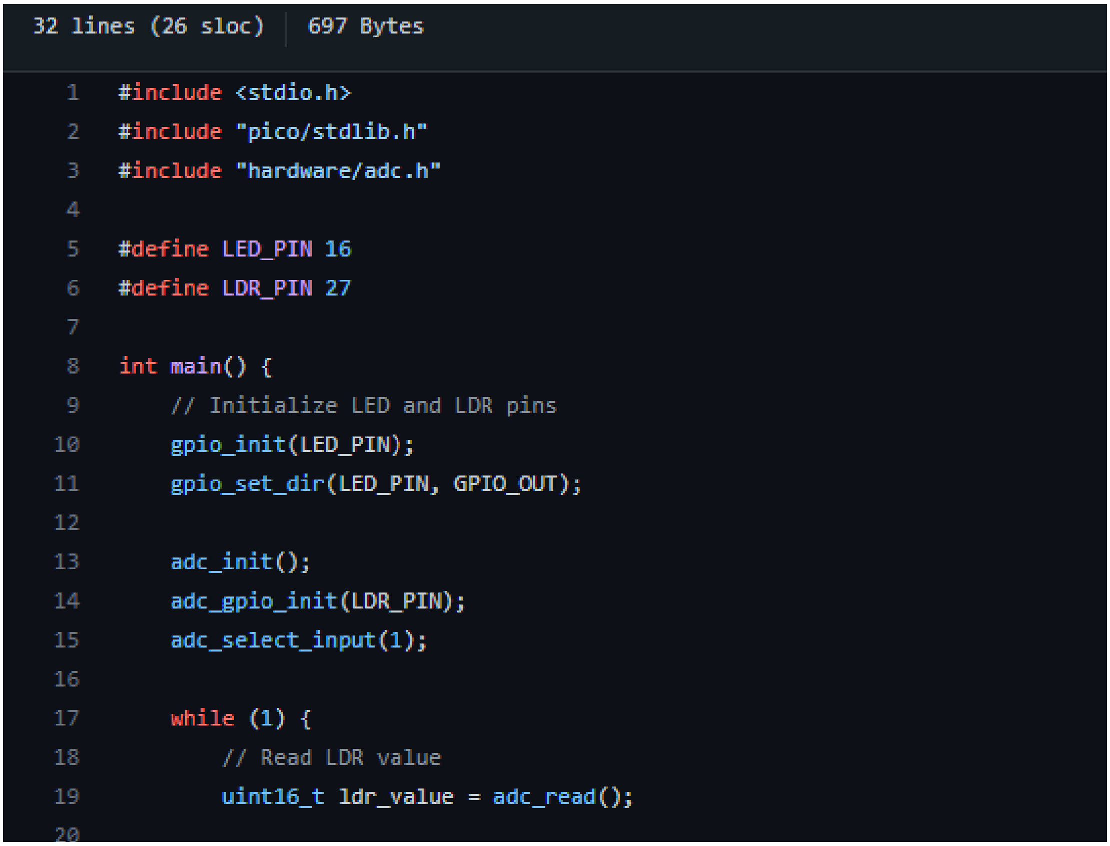


SCHEMATIC DIAGRAM



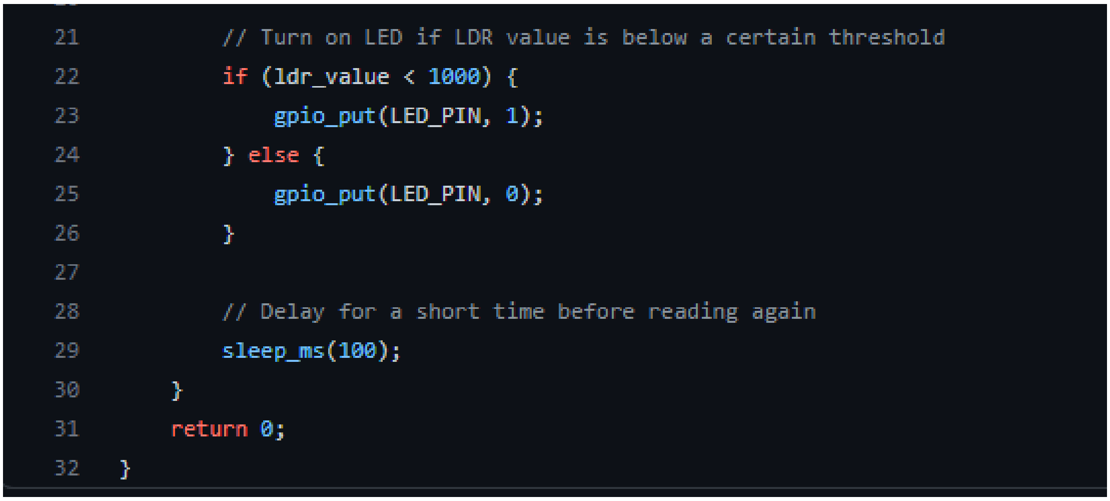


## Interfacing Raspberry Pi Pico with LDR and LED code in C/C++ Pico SDK



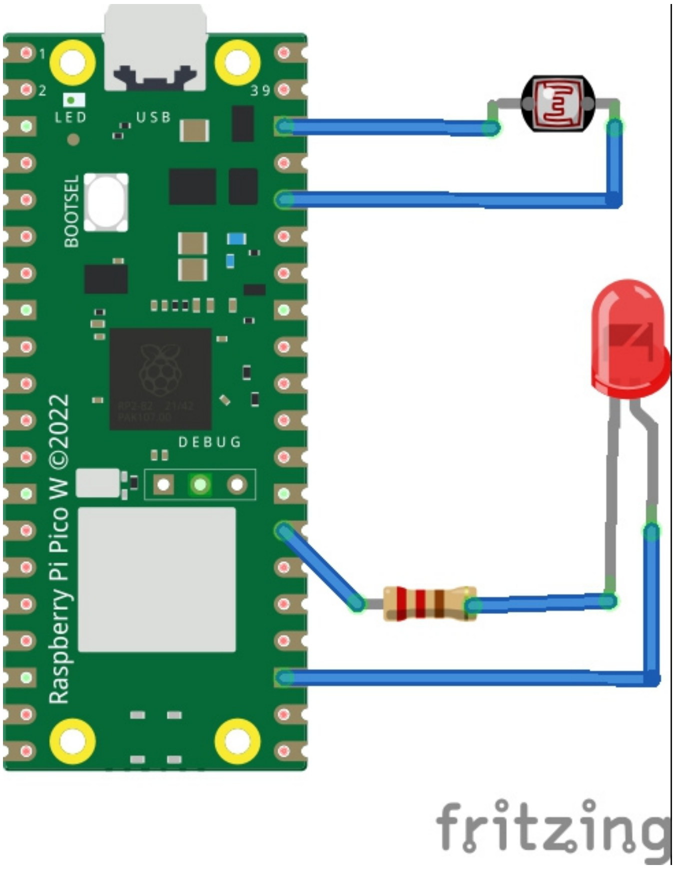
32 lines (26 sloc) | 697 Bytes

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

#define LED_PIN 16
#define LDR_PIN 27

int main() {
    // Initialize LED and LDR pins
    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);

    adc_init();
    adc_gpio_init(LDR_PIN);
    adc_select_input(1);

    while (1) {
        // Read LDR value
        uint16_t ldr_value = adc_read();

```

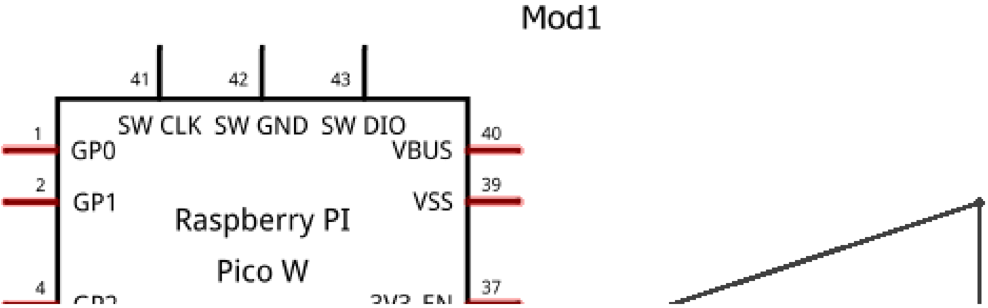


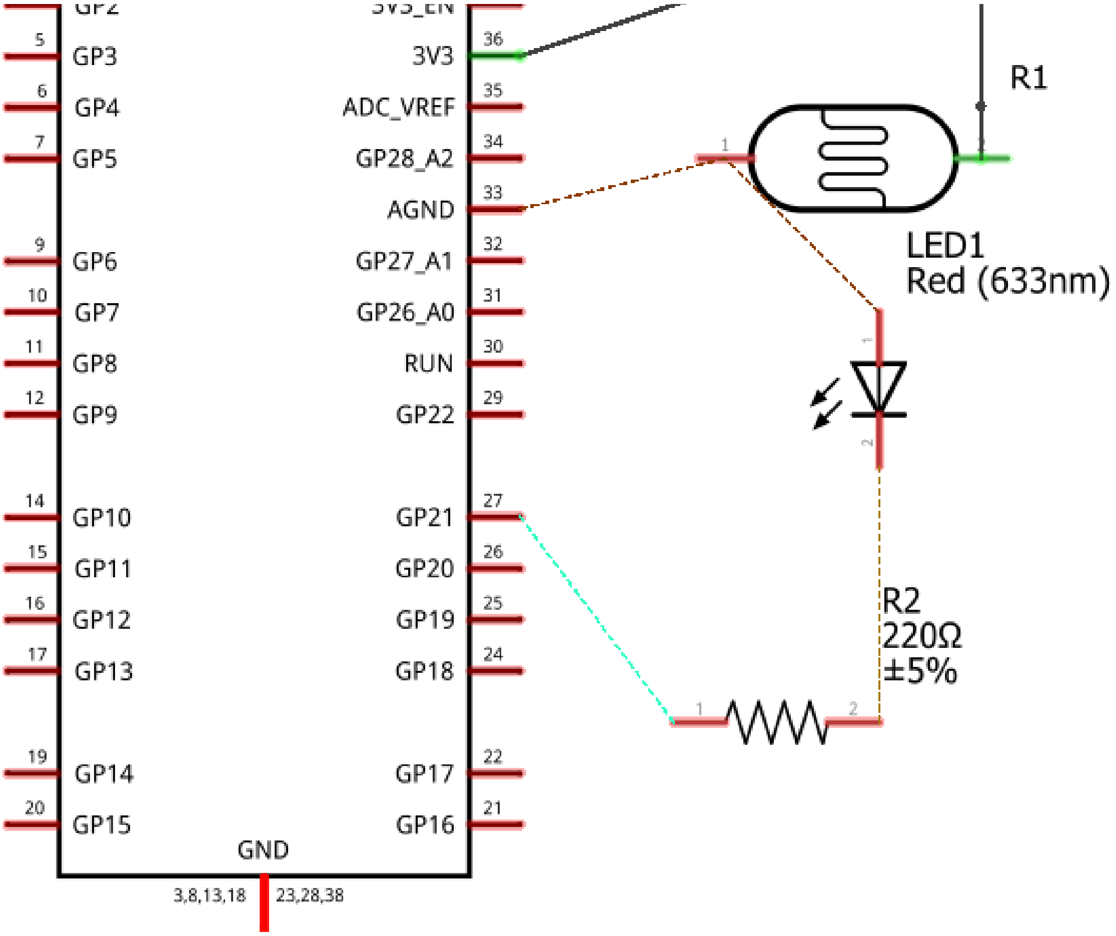
```c
        // Turn on LED if LDR value is below a certain threshold
        if (ldr_value < 1000) {
            gpio_put(LED_PIN, 1);
        } else {
            gpio_put(LED_PIN, 0);
        }

        // Delay for a short time before reading again
        sleep_ms(100);
    }
    return 0;
}
```

## Circuit Diagram :-



SCHEMATIC DIAGRAM





fritzing

# Thank you!