

```

#include <Wire.h>

const int buttonPin = D1; // Pin where the button is connected
const int ledPin = D2;    // Pin where the LED is connected

int buttonState = 1;      // Variable for reading the button status

void setup() {
  pinMode(buttonPin, INPUT_PULLUP); // Initialize the button pin as an input
  pinMode(ledPin, OUTPUT);          // Initialize the LED pin as an output
}

void loop() {
  // Read the state of the button
  buttonState = digitalRead(buttonPin);

  // Check if the button is pressed
  if (digitalRead(buttonPin) == HIGH) {
    // Turn the LED on
    digitalWrite(ledPin, LOW);
  }
  else {
    // Turn the LED off
    digitalWrite(ledPin, HIGH);
  }
}

```

Copilot said:

Copilot

It looks like you're working on a simple button-controlled LED project using an Arduino. Your code is almost correct, but there are a couple of things to note:

1. **Button State Logic:** The button state logic seems inverted. Typically, when using INPUT_PULLUP, the button reads LOW when pressed and HIGH when not pressed. So, you might want to adjust the logic in your loop() function.
2. **Initial Button State:** The initial state of buttonState is set to 1, but it's not used in the logic. You can remove it or use it appropriately.

Here's a revised version of your code:

This should work as expected, turning the LED on when the button is pressed and off when it's released. Let me know if you have any questions or need further assistance!

You said:

breytu þessum kóða þannig að ljósið ledinu logi þegar þrýst er á rofan og slökni þegar þrýst er aftur á rofan

Copilot said:

Copilot

Til að breyta kóðanum þannig að LED ljósið logi þegar þrýst er á rofann og slökni þegar þrýst er aftur á rofann, getur þú notað breytu til að fylgjast með ástandi LED ljóssins og breyta því í hvert skipti sem rofinn er þrýstur. Hér er uppfærður kóði:

Þessi kóði mun kveikja á LED ljósinu þegar rofinn er þrýstur og slökkva á því þegar rofinn er þrýstur aftur. Láttu mig vita ef þú þarft frekari aðstoð!

You said:

```
#include <Wire.h>
```

```
const int buttonPin = D1; // Pin where the button is connected
```

```
const int ledPin = D2; // Pin where the LED is connected
```

```
bool ledState = LOW; // Variable to store the LED state
```

```
bool buttonState = HIGH; // Variable to store the button state
```

```
bool lastButtonState = HIGH; // Variable to store the last button state
```

```
void setup() {
```

```
  Serial.begin(115200); //speed of the communications
```

```
  //Begin Communication, no need for any address because this is the master
```

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

```
  pinMode(buttonPin, INPUT_PULLUP); // Initialize the button pin as an input
```

```

pinMode(ledPin, OUTPUT);    // Initialize the LED pin as an output
digitalWrite(ledPin, ledState); // Set the initial state of the LED
}

void loop() {
    // Read the state of the button
    buttonState = digitalRead(buttonPin);

    Wire.beginTransmission(8); //node number 1

    // Check if the button is pressed and released
    if (buttonState == LOW && lastButtonState == HIGH) {
        Wire.write(1);
        Serial.println("1");
        Serial.println("LED ON master");
        // Toggle the LED state
        ledState = !ledState;
        Wire.write(0);
        Serial.println("0");
        digitalWrite(ledPin, ledState);
    }
    Wire.endTransmission();

    // Save the current button state as the last button state
    lastButtonState = buttonState;
}

```

Copilot said:

Copilot

Það lítur út fyrir að þú sért að bæta við I2C samskiptum í kóðann þinn. Það eru nokkur atriði sem þarf að laga til að tryggja að LED ljósið virki rétt þegar rofinn er þrýstur og slepptur. Hér er uppfærður kóði sem ætti að virka eins og þú vilt:

Í þessum kóða er LED ljósið kveikt þegar rofinn er þrýstur og slökkt þegar rofinn er þrýstur aftur. LED ástandið er einnig sent yfir I2C samskiptin og prentað í Serial Monitor. Láttu mig vita ef þú þarft frekari aðstoð!

You said:

here is the slave code

Copilot said:

Copilot

Great! Please share the slave code, and I'll help you integrate it with the master code you provided. This way, we can ensure the communication between the master and slave devices works correctly.

You said:

```
#include <Wire.h>
```

```
// int d1 = 0;
```

```
const int buttonPin = 27; // Pin where the button is connected
```

```
const int ledPin = 28; // Pin where the LED is connected
```

```
bool ledState = LOW; // Variable to store the LED state
```

```
bool buttonState = HIGH; // Variable to store the button state
```

```
bool lastButtonState = HIGH; // Variable to store the last button state
```

```
void setup() {
```

```
  pinMode(buttonPin, INPUT_PULLUP); // Initialize the button pin as an input
```

```
  pinMode(ledPin, OUTPUT); // Initialize the LED pin as an output
```

```
  digitalWrite(ledPin, ledState); // Set the initial state of the LED
```

```
  Serial.begin(115200);
```

```
  Wire.begin(8); //change address for the nodes in my case I use 8 and 9.
```

```
  Wire.onReceive(recieveEvent);
```

```
}
```

```
void loop() {
```

```
  // Read the state of the button
```

```

buttonState = digitalRead(buttonPin);

// Check if the button is pressed and released
if (buttonState == LOW && lastButtonState == HIGH) {
    // Toggle the LED state
    ledState = !ledState;
    Serial.println("LED ON NODE 1");
    digitalWrite(ledPin, ledState);
    //Serial.println("LED ON NODE 1");
}

// Save the current button state as the last button state
lastButtonState = buttonState;
}

void recieveEvent(int howMany)
{
    while (Wire.available())
    {
        buttonState = Wire.read();
    }
}

```

Copilot said:

Copilot

Thanks for sharing the slave code! Let's integrate it with the master code to ensure proper communication between the two devices. Here is the combined setup:

Master Code

Slave Code

In this setup:

- The master device toggles the LED state and sends the state over I2C.
- The slave device receives the LED state and updates its LED accordingly.

Let me know if you need any further adjustments or have any questions!

Today

You said:

have this in english

Copilot said:

Copilot

Sure! Here are the master and slave codes translated into English:

Master Code

Slave Code

In this setup:

- The master device toggles the LED state and sends the state over I2C.
- The slave device receives the LED state and updates its LED accordingly.

Let me know if you need any further adjustments or have any questions!