

Aufgabe 1:

Um den LED Streifen zu testen, öffne das auf dem Desktop befindliche Programm LED-Test.ino mit der Arduino IDE. Um das Programm auf das Wemos Modul zu übertragen, musst du folgende Einstellungen unter „Werkzeuge“ übernehmen:

1a. Wähle eine Farbe und lasse den gesamten LED-Streifen in dieser Farbe leuchten.

1b. Wähle fünf verschiedene Farben und lasse die LEDs auf dem Streifen in verschiedenen Farben leuchten.

1c. Lasse den LED-Streifen blinken.

1d. Implementiere eine kurze Animation.

Tipps:

<https://github.com/FastLED/FastLED/wiki/Basic-usage>

<https://github.com/FastLED/FastLED/wiki/Controlling-leds>

<https://github.com/FastLED/FastLED/wiki/Pixel-reference>

Arduino IDE > Datei > Beispiele > FastLED

Automatische Formatierung	Strg+T
Sketch archivieren	
Kodierung korrigieren & neu laden	
Serieller Monitor	Strg+Umschalt+M
Serieller Plotter	Strg+Umschalt+L
WiFi101 Firmware Updater	
Blynk: Check for updates	
Blynk: Example Builder	
Blynk: Run USB script	
Board: "WeMos D1 R2 & mini"	▸
Flash Size: "4M (1M SPIFFS)"	▸
Debug port: "Disabled"	▸
Debug Level: "Keine"	▸
lwIP Variant: "v2 Lower Memory"	▸
CPU Frequency: "80 MHz"	▸
Upload Speed: "921600"	▸
Erase Flash: "Only Sketch"	▸
Port: "/dev/ttyUSB1"	▸
Boardinformationen holen	
Programmer: "AVRISP mkII"	▸
Bootloader brennen	

Aufgabe 2:

Auf dem Desktop befindet sich die Datei blueprint.ino. Dieses Programm enthält die Grundfunktionalitäten um das Wemos-Modul mit einem WiFi-Netzwerk zu verbinden und Daten über das Protokoll MQTT zu empfangen.

Öffne die Datei mit der Arduino IDE. Die Kommentare helfen dir, dich im Programm zurecht zu finden.

Das Ziel ist es, den LED-Streifen mit einem Handy entweder lokal im eigenen Netzwerk oder über das Internet zu steuern. Dafür könnt ihr zum Beispiel IoT-MQTT Dashboard verwenden.

Bevor man mit dem Programmieren anfangen kann, muss man jedoch die Schnittstellen zwischen App und Mikrocontroller definieren.

2a. Definiere deine eigene Netzwerkschnittstelle für drei Eingabeparameter des Mikrocontroller. Stelle dir daher folgende Fragen: Welche MQTT-Topics werden wofür verwendet? Wie sieht der MQTT-Payload aus: welche Datentypen werden verwendet und wie soll der Mikrocontroller die einkommenden Daten reagieren?

2b. Implementiere deine Netzwerkschnittstelle.

Tipps:

<https://www.arduino.cc/reference/en/>