

ESP8266

Connect your Thing

Arduino IDE 1.6.5

Datei > Voreinstellungen:

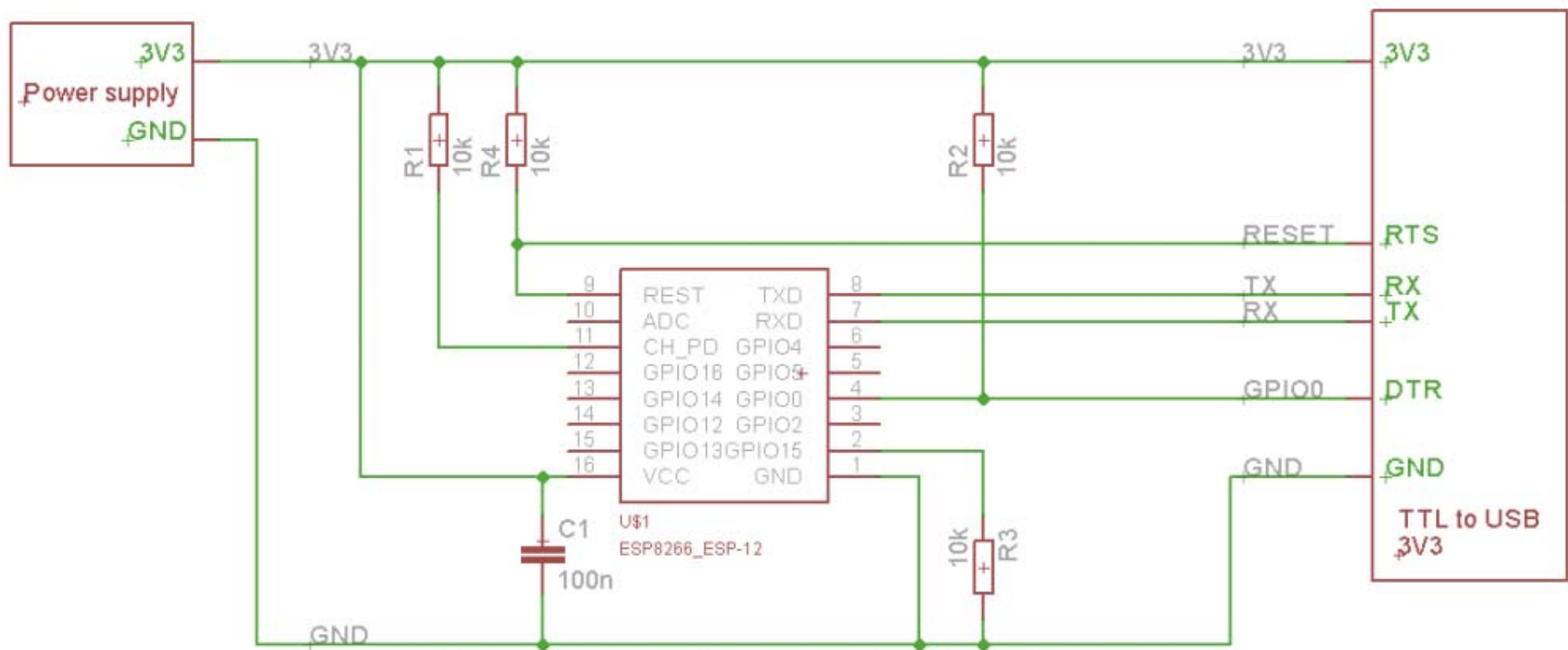
-> Additional Boards Manager URL

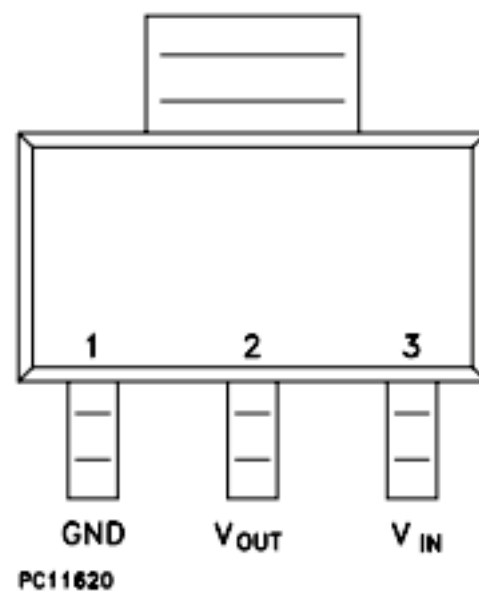
http://arduino.esp8266.com/package_esp8266com_index.json

Werkzeuge > Platine > Boards

Manager:

-> ESP8266 Installieren





SOT-223

```
//Digital I/O
```

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("hello world");  
  pinMode(12, OUTPUT);  
}
```

```
void loop() {  
  Serial.println("hello again");  
  delay(500);  
  digitalWrite(12, LOW);  
  delay(500);  
  digitalWrite(12, HIGH);  
}
```

GPIO	Inst Name	Function 0	Function 1	Function 2	Function 3	Function 4	At Reset	After Reset	Sleep
0	GPIO0 U	GPIO0	SPICS2			CLK OUT	oe=0, wpu	wpu	oe=0
1	U0TXD U	U0TXD	SPICS1		GPIO1	CLK RTC	oe=0, wpu	wpu	oe=0
2	GPIO2 U	GPIO2	I2SO WS	U1TXD		U0TXD	oe=0, wpu	wpu	oe=0
3	U0RXD U	U0RXD	I2SO DATA		GPIO3	CLK XTAL	oe=0, wpu	wpu	oe=0
4	GPIO4 U	GPIO4	CLK XTAL				oe=0		oe=0
5	GPIO5 U	GPIO5	CLK RTC				oe=0		oe=0
6	SD CLK U	SD CLK	SPICLK		GPIO6	U1CTS	oe=0		oe=0
7	SD DATA0 U	SD DATA0	SPIQ		GPIO7	U1TXD	oe=0		oe=0
8	SD DATA1 U	SD DATA1	SPID		GPIO8	U1RXD	oe=0		oe=0
9	SD DATA2 U	SD DATA2	SPIHD		GPIO9	HSPIHD	oe=0		oe=0
10	SD DATA3 U	SD DATA3	SPIWP		GPIO10	HSPIWP	oe=0		oe=0
11	SD CMD U	SD CMD	SPICS0		GPIO11	U1RTS	oe=0		oe=0
12	MTDI U	MTDI	I2SI DATA	HSPIQ MISO	GPIO12	U0DTR	oe=0, wpu	wpu	oe=0
13	MTCK U	MTCK	I2SI BCK	HSPIID MOSI	GPIO13	U0CTS	oe=0, wpu	wpu	oe=0
14	MTMS U	MTMS	I2SI WS	HSPICLK	GPIO14	U0DSR	oe=0, wpu	wpu	oe=0
15	MTDO U	MTDO	I2SO BCK	HSPICS	GPIO15	U0RTS	oe=0, wpu	wpu	oe=0
16	XPD_DCDC	XPD_DCDC	RTC_GPIO0	EXT_WAKEUP	DEEPSLEEP	BT_XTAL_EN	oe=1,wpd	oe=1,wpd	oe=1

http://www.esp8266.com/wiki/doku.php?id=esp8266_gpio_pin_allocations

Oder Google:

“ESP8266 Community Wiki”

```
//PWM
```

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("hello world");  
  pinMode(12, OUTPUT);  
}
```

```
void loop() {  
  Serial.println("hello again");  
  delay(500);  
  analogWrite(12, LOW);  
  delay(500);  
  analogWrite(12, HIGH);  
}
```

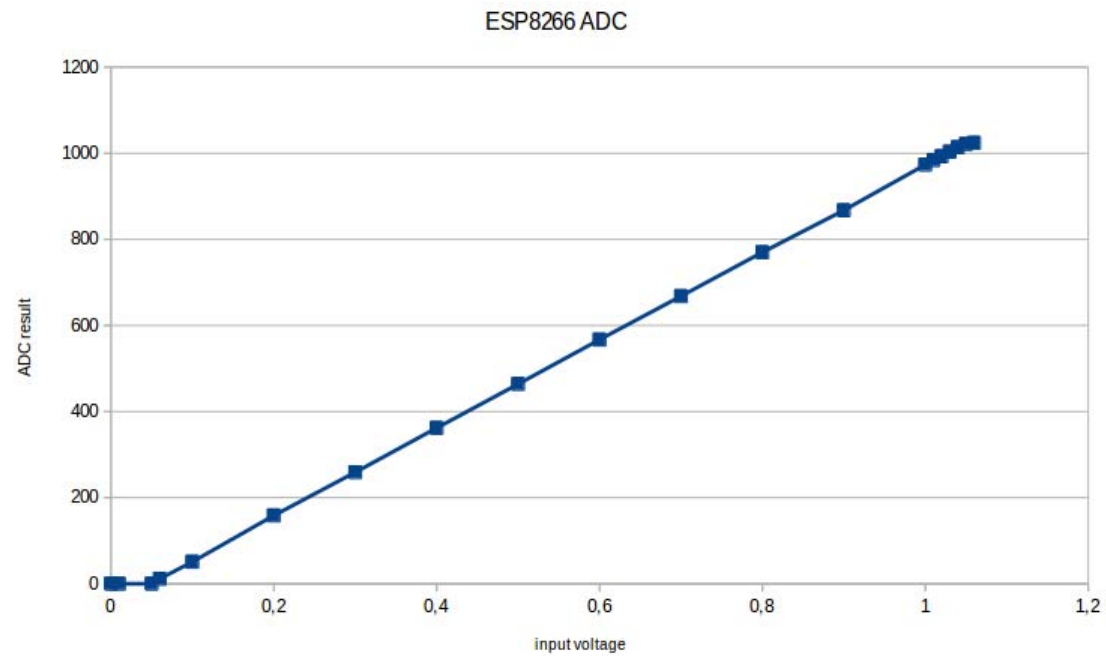
```
//analogRead
```

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("hello world");  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
  delay(500);  
}
```


Tabelle1

Voltage	ADC min	ADC max	ADC mean
0	0	0	0
0,01	0	0	0
0,05	0	0	0
0,06	9	14	11,5
0,1	49	53	51
0,2	157	160	158,5
0,3	257	261	259
0,4	360	364	362
0,5	463	465	464
0,6	565	569	567
0,7	666	670	668
0,8	768	772	770
0,9	865	870	867,5
1	971	976	973,5
1,01	982	986	984
1,02	990	996	993
1,03	1001	1006	1003,5
1,04	1011	1017	1014
1,05	1020	1023	1021,5
1,06	1024	1024	1024

Measurement equipment:
Labjack U12 (DAC output)



http://www.esp8266.com/wiki/doku.php?id=esp8266_gpio_pin_allocations

Oder Google:

“ESP8266 Community Wiki”

```
#include <ESP8266WiFi.h>

const char* ssid = "ssid";
const char* password = "password123";

void setup() {
  Serial.begin(115200);
  setup_wifi();
}

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {

}
```

MQTT

A little bit of history

MQTT was invented by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom, now Cirrus Link) back in 1999, when their use case was to create a protocol for minimal battery loss and minimal bandwidth [connecting oil pipelines over satellite connection](#). They specified the following goals, which the future protocol should have:

- Simple to implement
- Provide a Quality of Service Data Delivery
- Lightweight and Bandwidth Efficient
- Data Agnostic
- Continuous Session Awareness

These goals are still the core of MQTT, while the **focus has changed from proprietary embedded systems to open Internet of Things use cases**. Another thing that is often confused about MQTT is the appropriate meaning of the abbreviation MQTT. It's a long story, the [short answer](#) is that **MQTT officially does not have an acronym anymore, it's just MQTT**.

Real world applications [\[edit \]](#)

MQTT is a good choice for wireless networks that experience varying levels of latency due to occasional bandwidth constraints or unreliable connections.^[11] In the real world, there are a number of projects that implement MQTT.

- [Facebook Messenger](#). Facebook has used aspects of MQTT in Facebook Messenger.^[12] However, it is unclear how much of MQTT is used or for what; Moreover, it is to be noted that this is a phone application, not a sensor application.
- On October 8, 2015 [Amazon Web Services](#) announced Amazon IoT is based on MQTT.^[13]

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "ssid";
const char* password = "password123";
const char* mqtt_server = "broker.mqttdashboard.com";
const char* outTopic = "outTopic1";
const char* inTopic = "inTopic1";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[75];
int value = 0;

long lastReconnectAttempt = 0;
```

```
void setup() {  
  Serial.begin(115200);  
  setup_wifi();  
  client.setServer(mqtt_server, 1883);  
  client.setCallback(callback);  
}
```

```

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();

    if ((char)payload[0] == '1') {
        Serial.println("char 0 == 1");
        switch ((char)payload[1]) {
            case '0':
                Serial.println("Request 10");
                client.publish(outTopic, "request 10");
                break;
            case '1':
                Serial.println("Request 11");
                client.publish(outTopic, "request 11");
                break;
            case '2':
                Serial.println("Request 12");
                client.publish(outTopic, "request 12");
                break;
        }
    } else {

    }
}

```

```
boolean reconnect() {  
  Serial.println("reconnect()");  
  if (client.connect("arduinoClient")) {  
  
    client.publish(outTopic, "hello world");  
  
    client.subscribe(inTopic);  
  }  
  return client.connected();  
}
```

```

void loop() {

    long now = millis();

    if (!client.connected()) {
        if (now - lastReconnectAttempt > 5000) {
            lastReconnectAttempt = now;
            // Attempt to reconnect
            if (reconnect()) {
                lastReconnectAttempt = 0;
            }
        }
    }
    else {
        // Client connected
        if (now - lastMsg > 2000) {

            lastMsg = now;
            ++value;

            snprintf(msg, 75, "Hello nr. %d", value);
            Serial.print("Publish message: ");
            Serial.println(msg);
            client.publish(outTopic, msg);

        }
        client.loop();
    }
}

```