PWM, Wi-Fi and TCP Server

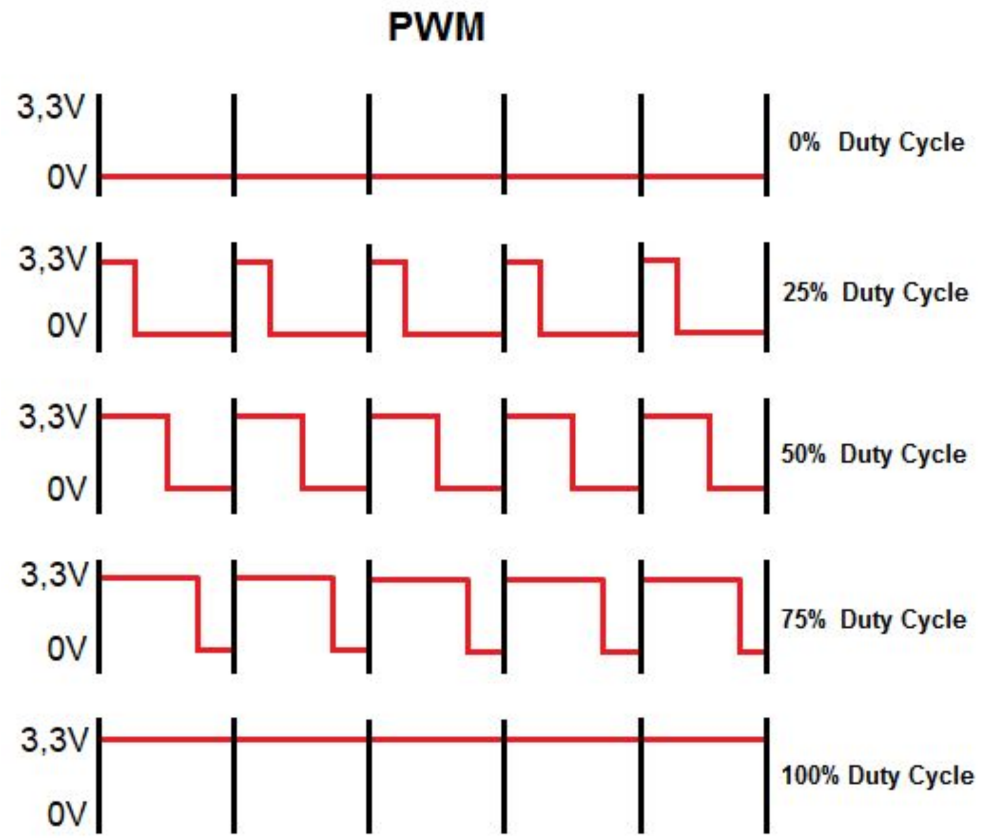
Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação
Tópicos Especiais em Sistemas de Informação - IoT

What is it?

Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of controlling the average power delivered by an electrical signal.

The average value of voltage (and current) fed to the load is controlled by switching the supply between 0 and 100% at a rate faster than it takes the load to change significantly.

PWM and Duty Cycle



PWM and Arduino

Use any digital pin to connect a PWM load.

The function analogWrite is used to send the duty cycle to a pin.

```
analogWrite( pin, value)
```

where,

pin: any digital pin

value: [0, 1023] (10 bits)

PWM and Arduino

```
int pwm;

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(115200);
}
```

```
void loop()
{
        analogWrite( LED_BUILTIN, pwm );

        Serial.print("O valor do pwm é: ");
        Serial.println(pwm);
        pwm = pwm +100;
        delay(1500);

        if( pwm > 1023 )  {
             pwm = 0;
        }
}
```

Station Mode

```
#include <ESP8266WiFi.h>

const char* ssid     = "SSID";
const char* password = "PASSWORD";

void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println('\n');

  WiFi.begin(ssid, password);
  Serial.print("Connecting to ");
  Serial.print(ssid); Serial.println(" ...");

  int i = 0;
  while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.print(++i); Serial.print(' ');
  }

  Serial.println('\n');
  Serial.println("Connection established!");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());
}

void loop() { }
```

Tarefa

Desenvolver um serviço Web que receba o valor do duty cycle aplicado em um pino do ESP8266/ESP32.

- Desenvolver um servidor TCP no computador
    - Exemplo: https://realpython.com/python-sockets/
- Aplicar o PWM ao pino LED_BUILTIN
- Criar um cliente Web no Arduino para se conectar ao servidor criado acima
    - Usar o exemplo do WiFiClientBasic existente na IDE do Arduino
- Enviar os dados do duty cycle para o servidor.

Wi-Fi to Router

```
#include <ESP8266WiFi.h>

#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

const char* host = "djxmmx.net";
const uint16_t port = 17;
```

```
void setup() {
  Serial.begin(115200);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  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() {
  static bool wait = false;

  Serial.print("connecting to ");
  Serial.print(host);
  Serial.print(':');
  Serial.println(port);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  if (!client.connect(host, port)) {
    Serial.println("connection failed");
    delay(5000);
    return;
  }
```

```
// This will send a string to the server
Serial.println("sending data to server");
if (client.connected()) { client.println("hello from ESP8266"); }

// wait for data to be available
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    client.stop();
    delay(60000);
    return;
  }
}
```

```
  // Read all the lines of the reply from server and print them to Serial
  Serial.println("receiving from remote server");
  // not testing 'client.connected()' since we do not need to send data here
  while (client.available()) {
    char ch = static_cast<char>(client.read());
    Serial.print(ch);
  }

  // Close the connection
  Serial.println();
  Serial.println("closing connection");
  client.stop();

  if (wait) {
    delay(300000);  // execute once every 5 minutes, don't flood remote service
  }
  wait = true;
}
```

Wi-Fi WPA2

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

#include "wpa2_enterprise.h"

char ssid[] = "UFPA 2.0 - Institucional";
char username[] = "dionne@ufpa.br";
char password[] = "";
```

```
void setup()
{
  Serial.begin(115200);
  wifi_set_opmode(STATION_MODE);

  // Configure SSID
  struct station_config wifi_config;
  memset(&wifi_config, 0, sizeof(wifi_config));
  strcpy((char *)wifi_config.ssid, ssid);
  wifi_station_set_config(&wifi_config);

  // DO NOT use authentication using certificates
  wifi_station_clear_cert_key();
  wifi_station_clear_enterprise_ca_cert();
```

```
// Authenticate using username/password
wifi_station_set_wpa2_enterprise_auth(1);
wifi_station_set_enterprise_identity((uint8 *)username, strlen(username));
wifi_station_set_enterprise_username((uint8 *)username, strlen(username));
wifi_station_set_enterprise_password((uint8 *)password, strlen(password));

// Connect
wifi_station_connect();

// Wait for connect
while (WiFi.status() != WL_CONNECTED)
{
  Serial.println("Wifi connecting...");
  delay(500);
}
```

```
  // Print wifi IP addess
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```