**Assembly of circuit with speed variation (PWM) water sensor with for car and sending IoT data**

**Difficulty level:** **Difficult**

**Goals**

Automotive IoT is the integration of gadgets, sensors, cloud computing, applications, and other such components into vehicles to function as a complex system for the connection of cars, predictive maintenance, fleet management, OEMs, insurance, and more. The integration of the Internet of Things in the automotive industry allows manufacturers to implement sought-after innovations that can ultimately transform cars into near-artificial intelligence. At a didactic level, we are now going to develop some exercises using sensors for data acquisition, processed by the Arduino microcontroller.

This exercise can be applied when weather conditions provide rainy weather and thus water can be detected on the vehicle's windows helping to clean autonomously, controlling the cleaning speed of the water.

For the possible sending of data, it will be necessary to apply, for example, the ESP8266 ESP-01 module that allows the connection of several devices to the internet (or local network), and consequent sending of data from the sensors applied to the autonomous system.

**Image-1:** Understanding the application of sensors in a car and communicating with IoT.

Uma imagem com texto, captura de ecrã, design

Descrição gerada automaticamente

**Image 1:** application of water sensor in a car and communicating with IoT

**Skills**

* The skills our students will gain are:
* Students' ability to build circuits will be developed.
* The ability to program the Arduino board and use the ESP8266 Module for Internet access will develop.
* The ability to receive data from the brightness sensor and send the received data to Thing Speak will be gained.
* Data analytics will improve their ability to connect with the Internet of Things.

**Required materials and circuit diagram.**

In this exercise we intend to learn how to draw diagrams (circuits), connect all the components correctly, develop software based on C language (Arduino), connect to the wifi network, communicate with an IoT server, ThingSpeak and read server-generated graphics.

|  |  |
| --- | --- |
| **Quantity** | **Component** |
| 1 | Arduino Uno R3 |
| 1 | ESP01-8266 |
| 1 | Power Supply (braedBoard) |
| 1 | BreadBoard |
| 1 | Water sensor. |
| 1 | DC Motor |
| 1 | Module bridge L298N |

**Table 1 - Components List**

**Materials table**

|  |  |
| --- | --- |
| Arduino | ESP01 - 8266 |
| Bread Board + Power Supply | Uma imagem com ferramenta  Descrição gerada automaticamente  Water Sensor |
| DC Motor | L298N Bridge Motor Module |
| Jumper wire | |

Uma imagem com Engenharia eletrónica, máquina, eletrónica, ferramenta

Descrição gerada automaticamente

**Image 2 – Diagram circuit**

**Implementation**

Development of communication of microcontroller systems, and sensors, with the ThingSpeak IoT cloud.

The ESP8266 WiFi module (image 3) is a small shield with integrated TCP/IP protocol that can give any microcontroller access to the WiFi network. The ESP8266 is capable of both hosting an application and offloading all WiFi network functions from another application processor. Each ESP8266 module is pre-programmed with an AT command making its firmware settings, meaning that we can simply connect this module to the Arduino working as any other WiFi shield would. This module has a great cost/benefit ratio and has a very large and constantly growing user community.

Uma imagem com texto, eletrónica, circuito

Descrição gerada automaticamente

**Image 3 - ESP01 – 8266**

The module water sensor (image 4) is easy- to-use, portable and cost-effective, it is designed to identify and detect water level and water drop. This sensor measures the volume of water drop and water quantity through an array of traces of exposed parallel wires. Compared with its competitors, this sensor is not only smaller and smarter but also ingeniously equipped with following features:

Smooth conversion between water quantity and analogy quantity.

Strong flexibility, this sensor outputs basic analogy value.

Low power consumption and high sensitivity.

Directly connected to microprocessor or other logic circuits, suitable for a variety of development boards and controllers such as Arduino controller, STC single-chip microcomputer, AVR single-chip microcomputer.

Uma imagem com ferramenta

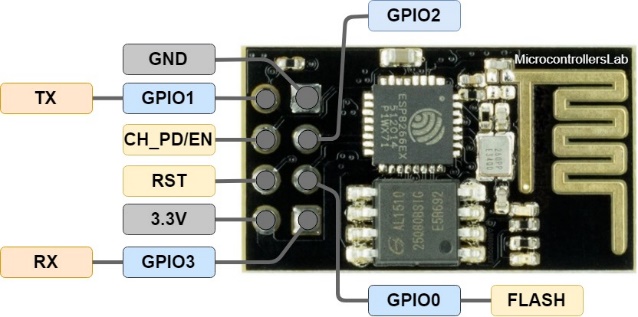
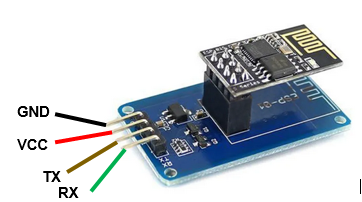
Descrição gerada automaticamente

**Image 4 Water Sensor**

**Implementation in practice**

1. Assemble the circuit in the image 2;
2. Connect correctly ESP01-8266 image 5

**Image 5** ESP-01 Connections



1. Real assembled circuit image 6

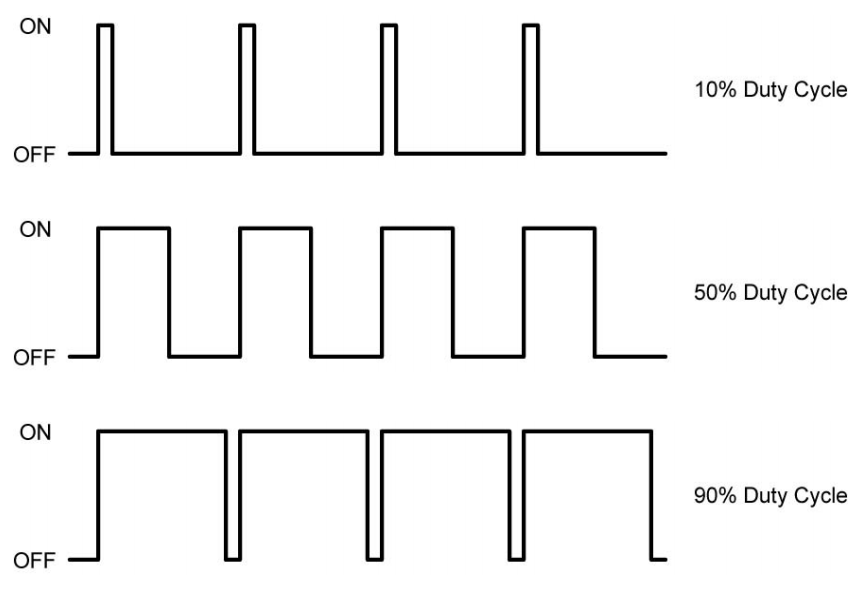
Uma imagem com eletrónica, Engenharia eletrónica, Fios elétricos, Componente de circuito

Descrição gerada automaticamente

**Image 6** Real circuit in breadboard

1. PWM (Pulse Width Modulation)

The PWM can be implemented in several areas of electronics. One of its uses is in power supply, DC motors speed control, light control, servo motors control and several other applications. Through PWM we can control speed and power.



**PWM**

**PWM**

**Period**

**Picture 1:PWM**

## **PWM operation**

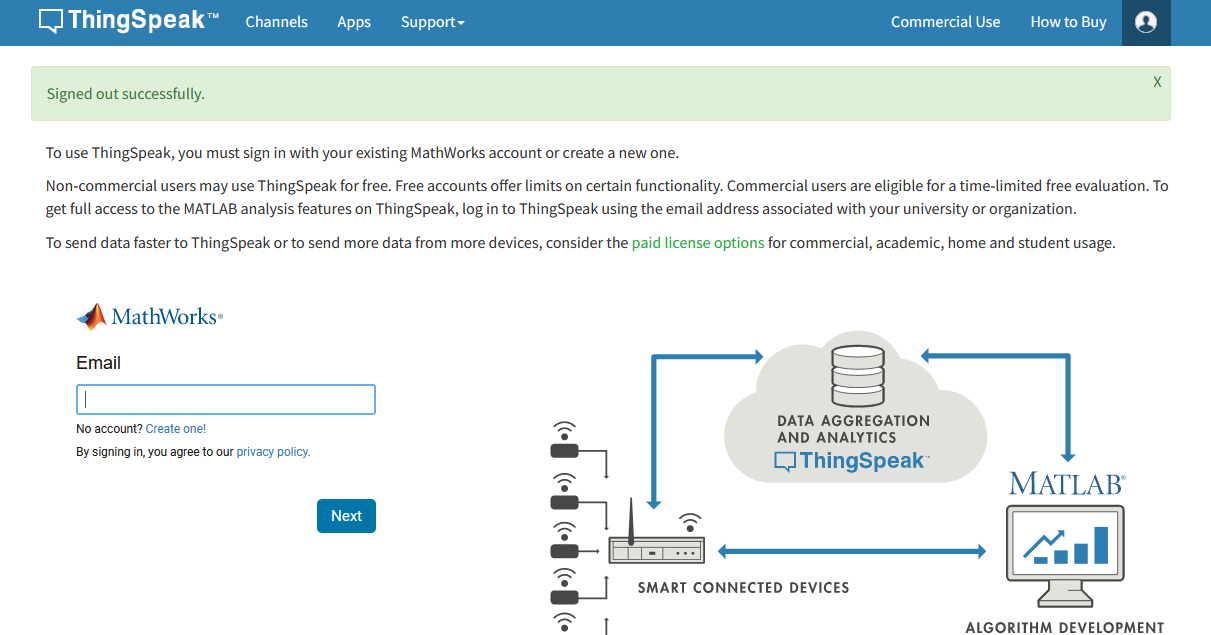
Considering a square wave, to obtain the correct operation of the PWM we must vary the pulse width of the wave. To calculate we need the period and the pulse width and its result is called duty-cycle, as it is defined by the equation:

**Duty cycle:** percentage value;

**Pulse Width:** sequence of time in which the signal is in high level;

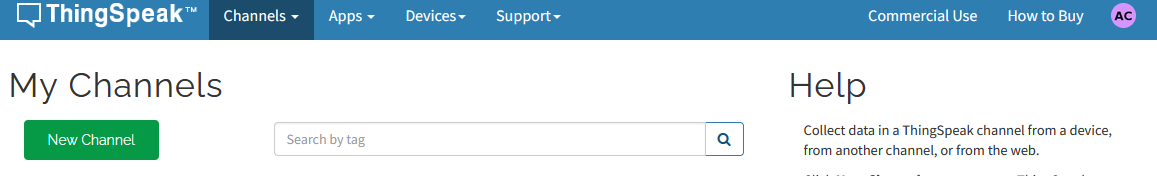
**Period:** duration of a wave cycle.

1. Create a ThingSpeak account image 7



**Image 7 - Thing Speak**

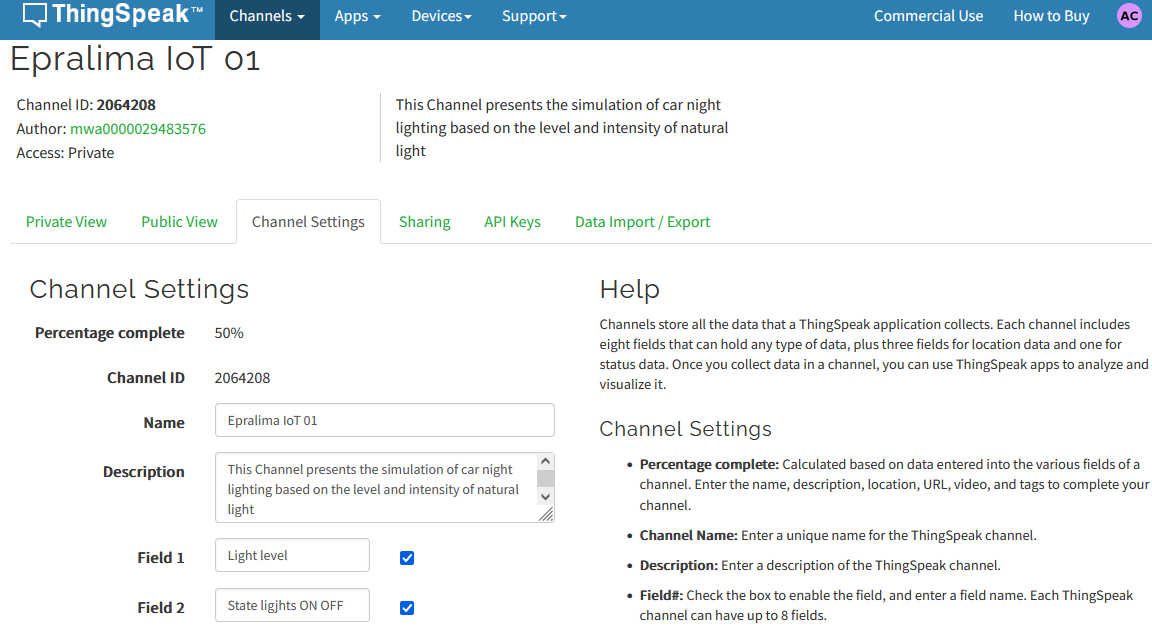
1. Create a new channel image 8



**Image 8 Interface ThingSpeak**

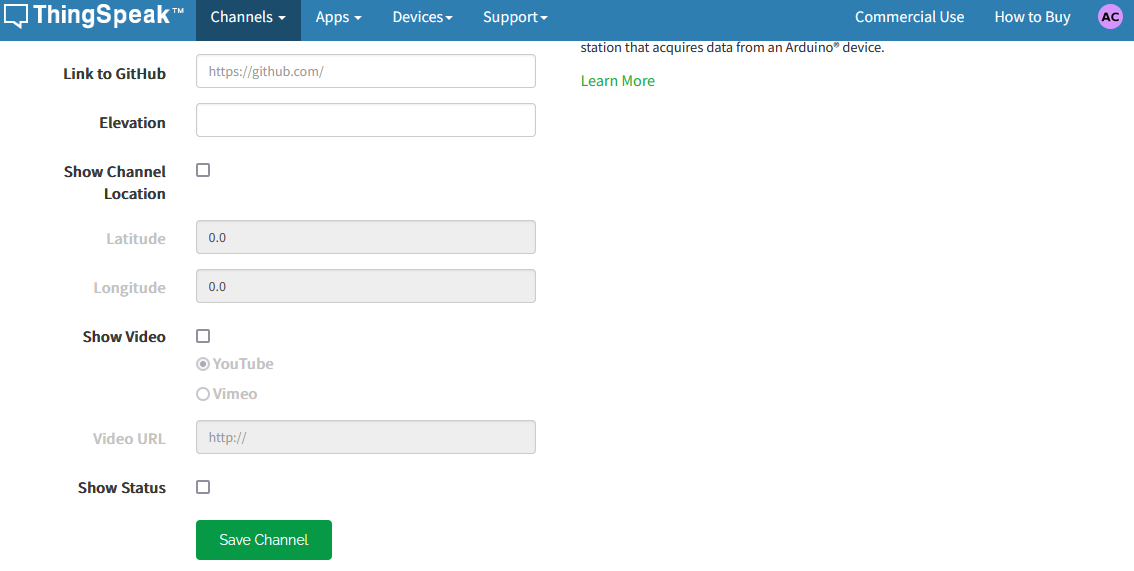
1. Configure channel, with name, description and fields. Image 9.

**Note:** The fields refer to data processed by the microcontroller and data from the sensors under study. Each field will generate a graph.



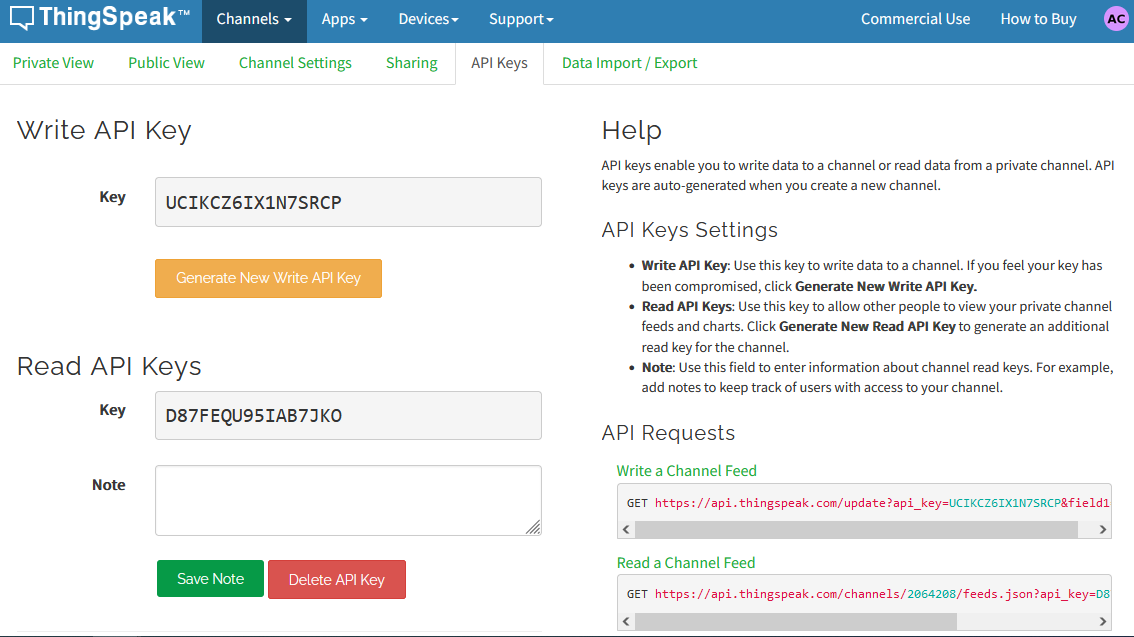
**Image 9 Configure Channel**

1. Save settings channel Image 10



**Image 10 Save settings channel**

1. In this step, we will pay special attention to the api keys, as they are the ones that, through the string key, will allow access to the IoT repository in Arduino programming. Also very important are the API requests.



**Image 11 - API Keys**

1. Programming Arduino

Inclusion of the necessary libraries and declaration of variables and constants inherent to the program's operation.

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Void setup() function for initializing parameters for starting the program.

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

**AT commands**

AT commands are the basic way to configure and trigger the ESP8266 when it is under control of an external device (like an Arduino, for example).

Current AT commands are direct descendants of the so-called "Hayes Standard" from 1981, used to allow personal computers to interact with telephone connections by directly controlling a mode.

The **InitWifiModule()** function initializes the ESP8266 through AT commands.

Uma imagem com texto

Descrição gerada automaticamente

The **envioDadosESP\_AT(str,int,boolean)** function is responsible for sending AT commands to the ESP8266

Uma imagem com texto

Descrição gerada automaticamente

The **startThingSpeakCmd(str,int,boolean)** function opens connection to ThingSpeak IoT analytics platform. The IP address of the ThingSpeak platform is: 184.106.153.149 with connection on port 80. The AT command to start ThingSpeak communication is AT+CIPSTART=PROTOCOL, IP\_ADRESS, PORT.

Uma imagem com texto

Descrição gerada automaticamente

The **EscreverParaThingSpeak** function generates a string to build an API Request.

**Example:**

**GET /update?api\_key=U………….P&field1= 0&field2= 0**

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

The **GetThingSpeak(str)** function, is responsible for determining and sending an API Request through the AT+CIPSEND command to write to the ThingSpeak channel, returning the message received by the response from the ThingSpeak data platform. The communication will be closed if the response is not favourable.

Uma imagem com texto

Descrição gerada automaticamente

The **SensorWaterRead()** function receives data from the water sensor on an analogy port (A0) on the Arduino. Depending on the amount of water, it turns the engine on or off.

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

**Results**

Considering that there is a certain amount of water on the front glass of the car, it is possible to automatically turn on the windshield wiper blades. It is possible to analyse the graph and observe a decrease in water and the turning on/off of the motor that activates the windshield wiper blades. With PWM (Pulse With Modulation) technology, it is possible to control the cleaning speed of the water depending on the amount of water.

Uma imagem com texto, file, diagrama, captura de ecrã

Descrição gerada automaticamente

**Image 12 – Results IoT ThingSpeak**

The data acquired by the ThingSpeak IoT platform can also be exported to CSV files and consequently imported into datasheets as shown in Table 2

Uma imagem com texto, captura de ecrã, número, menu

Descrição gerada automaticamente

**Tabela 2 - DataSheet**

**In short**

This can be applied when weather conditions provide rainy weather and thus water can be detected on the vehicle's windows helping to clean autonomously, controlling the cleaning speed of the water.