**Assembly of circuit with sensors Light Dependent Resistor for car and sending IoT data**

**Difficulty level:** Medium

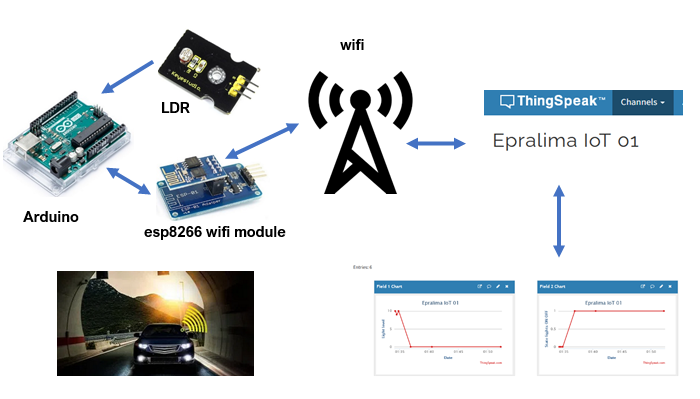
**Goals**

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 intends to apply light sensors guided by a microcontroller to autonomously control the night lighting of a vehicle in order to maintain the safety of the driver and occupants. 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.



**Image 1:** application of sensors 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.**

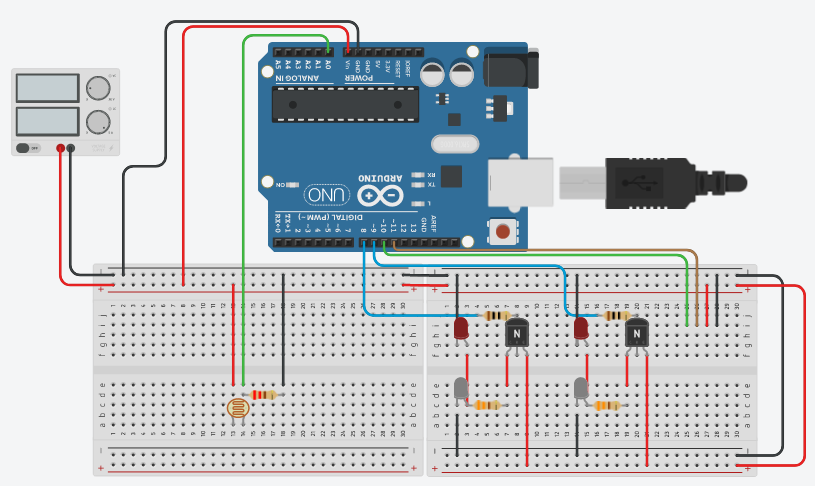
In this exercise we intend to learn how to draw diagrams (circuits), connect all the components correctly, develop software based on C language (Arduino), connect to the wifi network, communicate with an IoT server, ThingSpeak and read server-generated graphics.

|  |  |
| --- | --- |
| **Quantity** | **Component** |
| 1 | Arduino Uno R3 |
| 1 | ESP01-8266 |
| 1 | Power Supply (braedBoard) |
| 1 | BreadBoard |
| 1 | RFotoresistor. |
| 1 | 220 Ω Resistor |
| 2 | Transistor NPN (BJT) |
| 2 | 330 Ω Resistor |
| 2 | White LED |
| 2 | Red LED |
| 2 | 10 Ω Resistor |

**Table 1 - Components List**

**Materials table**

|  |  |
| --- | --- |
| Arduino | ESP01 - 8266 |
| Bread Board + Power Supply | LDR |
| 220Ω 330Ω 10Ω | White LED Red LED |
| Jumper wire | |



**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 LDR (image 4) is ideally suited to light detection, typically seen in project robots and daylight systems. Based on a light dependant resistor, a potential divider circuit provides an analogy output in response to the amount of light available and the DC input.

Working at a low 3.3 - 5V input the module will output high values on detecting light and low values while not detecting. Furthermore mounting holes ensure you have some method of mounting the unit to all manner of projects.

*Uma imagem com texto

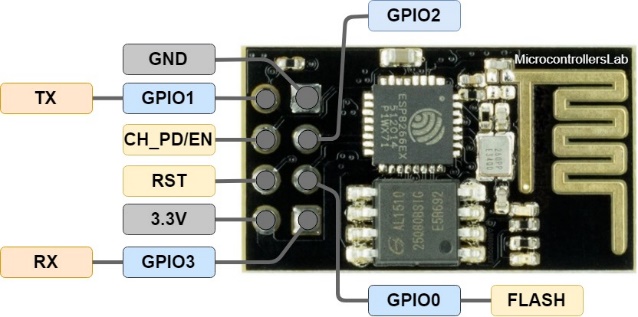
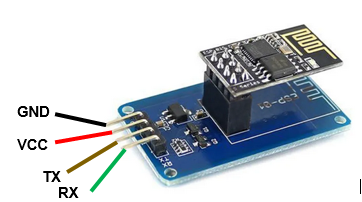
Descrição gerada automaticamente*

**Image 4 l LDR**

**Implementation in practice**

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

**Image 5** ESP-01 Connections



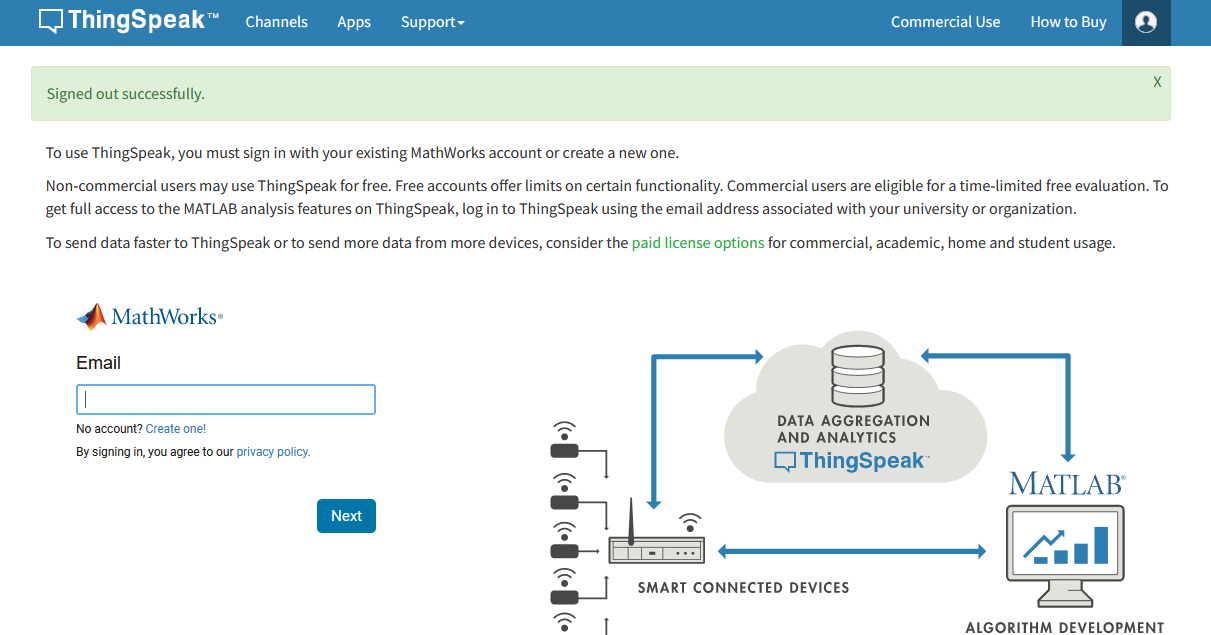
1. Real assembled circuit image 6

Uma imagem com texto

Descrição gerada automaticamente

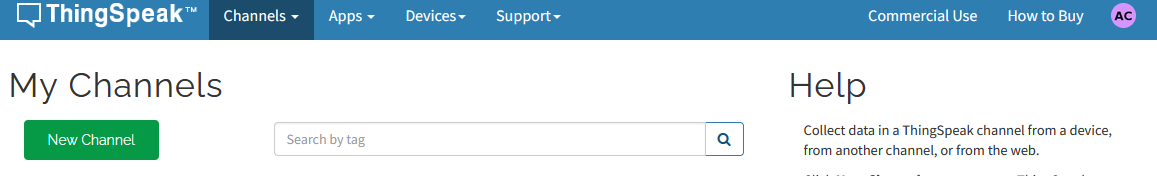
**Image 6** Real circuit in breadboard

1. Create a ThingSpeak account image 7



**Image 7 - Thing Speak**

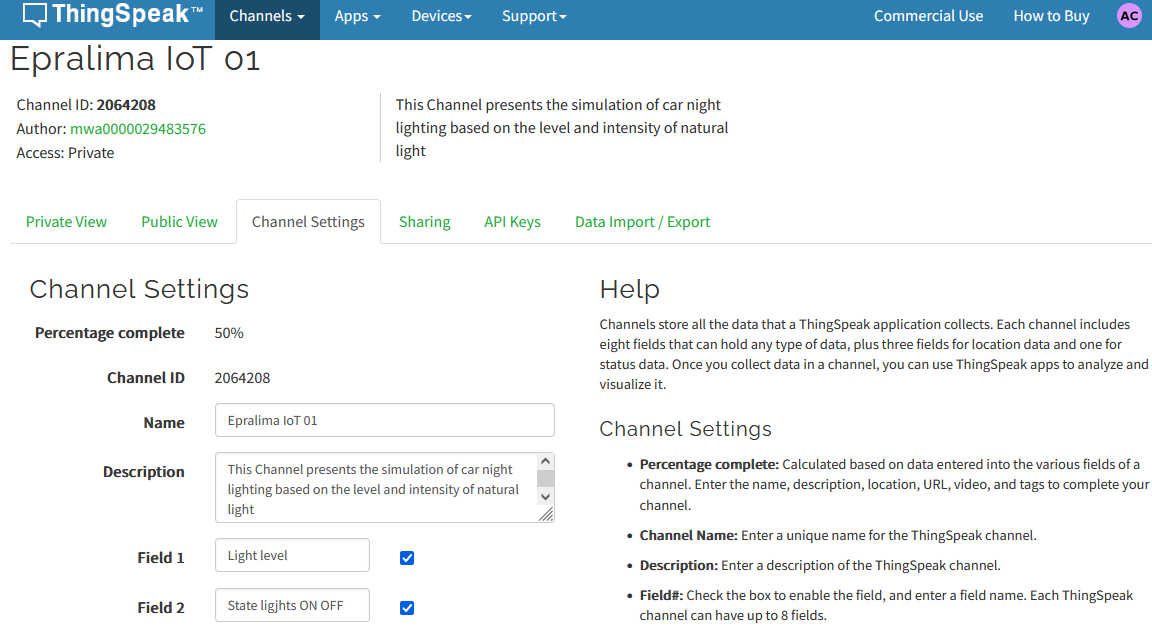
1. Create a new channel image 8



**Image 8 Interface ThingSpeak**

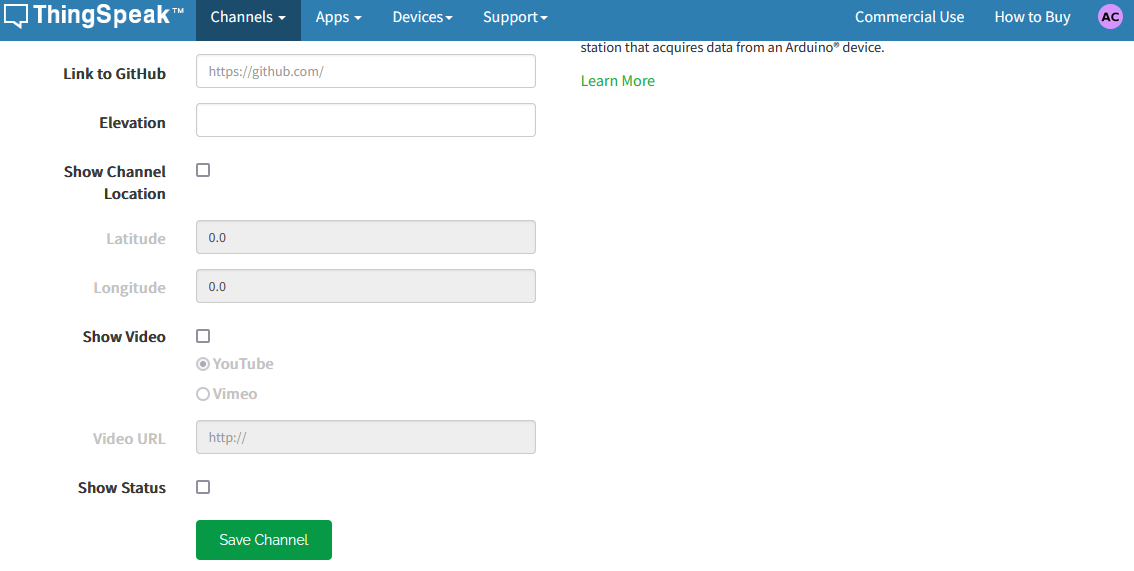
1. Configure channel, with name, description and fields. Image 9.

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



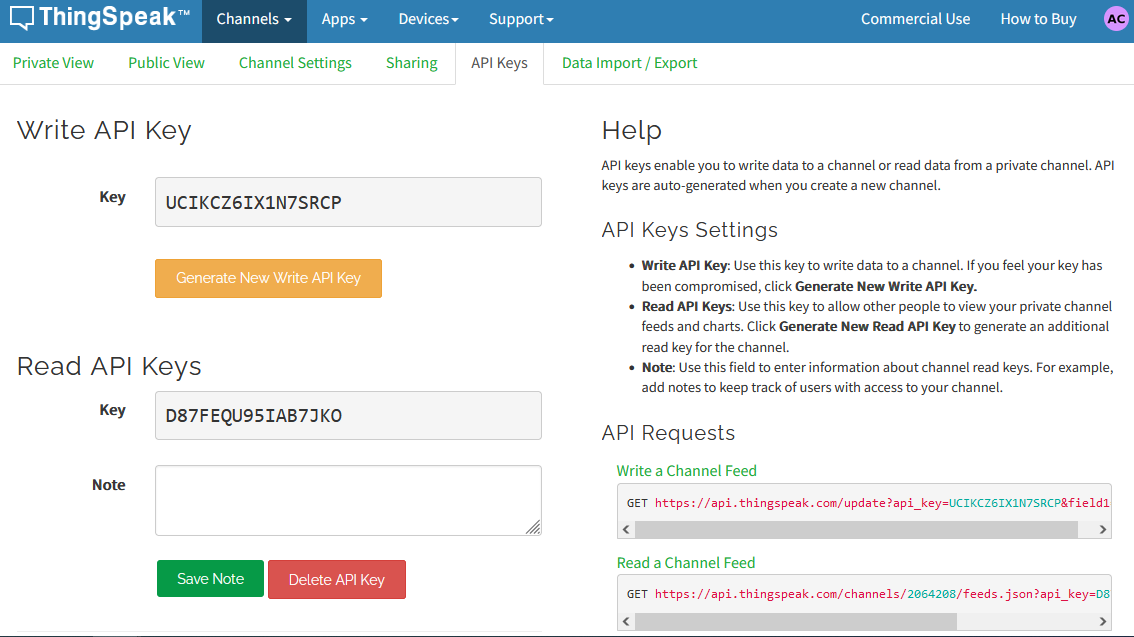
**Image 9 Configure Channel**

1. Save settings channel Image 10



**Image 10 Save settings channel.**

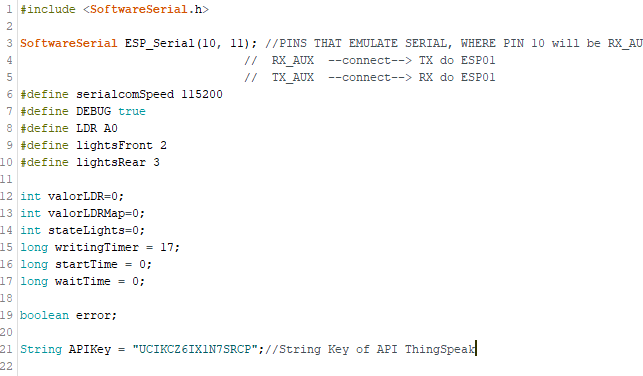
1. In this step, we will pay special attention to the api keys, as they are the ones that, through the string key, will allow access to the IoT repository in Arduino programming. Also very important are the API requests.



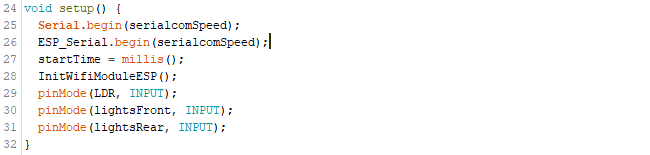
**Image 11 - API Keys**

1. Programming Arduino

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



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



**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.

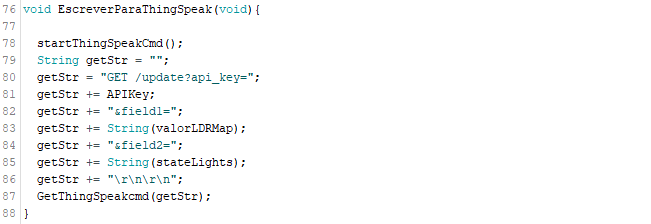
Uma imagem com texto

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**



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

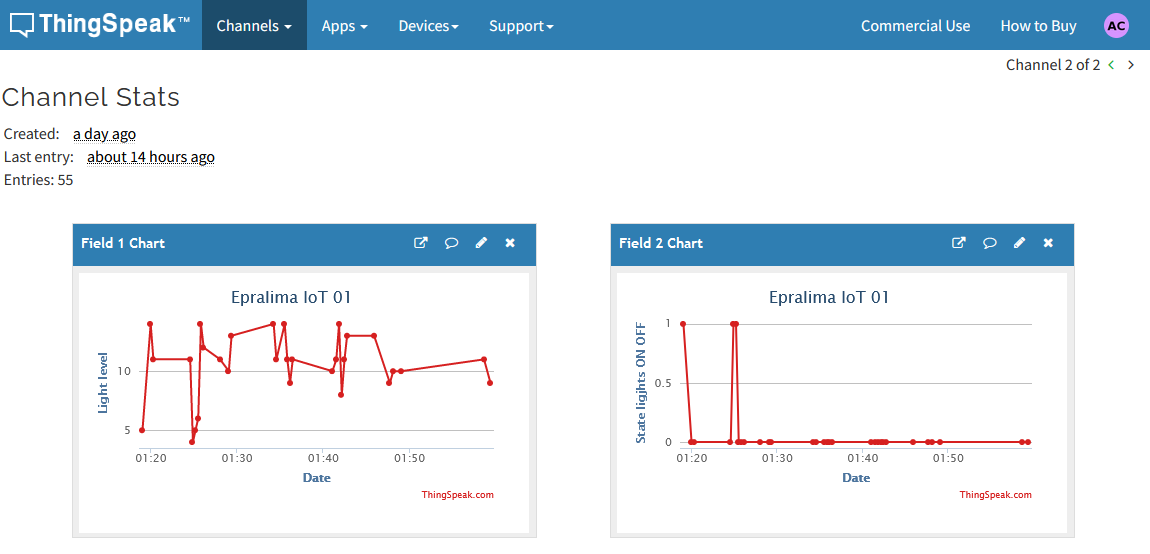
The **LerNivelIluminacao()** function receives data from the LDR sensor on an analogy port (A0) on the Arduino. Depending on the brightness setting, it turns on/off digital ports 2 and 3.

Uma imagem com texto

Descrição gerada automaticamente

**Results**

Considering that the night lighting of a car must turn on when the light level returned by the LDR is less than 5, it is possible to verify through the graphs generated by the online data analyser ThingSpea, that the light levels returned are mostly above 5 (Light Level). This implies that the night lighting lights are off for a longer time.



**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 mesa

Descrição gerada automaticamente

**Tabela 2 - DataSheet**

**In short**

If this were applied, we could conclude that the vehicle could be driving during the day and go through a dark area with little natural lighting, for example.