

INTELLIGENT CONNECTED OBJECTS

SUBJECT	Controlling a LED through an app
FORMAT	Individual, pairs, or small groups
PREPARATION TIME	1 hour
ACTIVITY LENGTH	1h30
DIFFICULTY LEVEL	Advanced

PEDAGOGICAL GOALS

- To develop an IoT project
- To work on implementing a sensor application
- To control the microcontroller through an application

NECESSARY MATERIALS Group 1

- Computer room with computers connected to the Internet
- 1 cell phone with Android OS per group
- NodeMCU board or similar
- Breadboard
- Jumpers
- LED
- 220R resistor

INTELLIGENT CONNECTED OBJECTS

Conducting the activity:

In this activity, students will control an LED using a mobile app built in AppInventor.

- If possible, divide the class into small groups. Each group must have access to a computer with Arduino IDE installed, a NodeMCU board or similar and a cable to connect the board to the computer. If this is not possible, try to take turns so that all students have access to the materials, or work in a collective format, projecting your screen.
- Organize the materials and provide a kit with the components for each group.
- Then ask each group to connect their board to their computer and open the Arduino IDE.
- Present the steps to transfer the code below onto the board.

Code

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

int reading;
int led = D7;

// Network name and password
char ssid[] = "#####"; // SSID SSID network name
char pass[] = "#####"; // Password

int status = WL_IDLE_STATUS;
WiFiClient client;
int sensorValue; // variable that stores the read analog value

// Information available on ThingSpeak, channel number and writing key
unsigned long channel = #####;
const char * ReadingKeyAPI = "#####";

void setup() {
    WiFi.begin(ssid, pass);
    ThingSpeak.begin(client);
    Serial.begin(115200);
    pinMode(led, OUTPUT);
}

void loop() {
    //Reads the last value written into the channel from the parameters below (CHANNEL, FIELD, Read Key)
    reading= ThingSpeak.readLongField(channel, 1, ReadingKeyAPI);

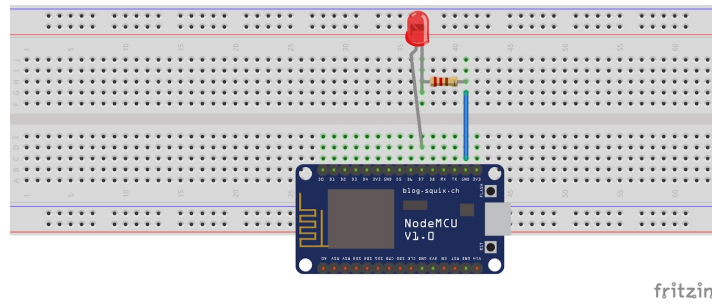
    //Writes the value read in Serial Monitor
    Serial.print("Reading: ");
    Serial.println(reading);

    if(reading == 1){
        digitalWrite(led, 1);
    }

    if(reading == 0){
        digitalWrite(led, 0);
    }
}
```

INTELLIGENT CONNECTED OBJECTS

- Check the project circuit below:



- Once Thingspeak's network parameters, password, write and read keys, and channel have been changed, ask them to update and test whether the project is working.

Controlling through an app

To control the project through an application, install the application found on this [link](#) or open this [project](#) in the AppInventor website, and use emulation to work with it. At first, update the writing key in the installed application and try to control the LED using IoT.

In a second step, use the AppInventor application project example as inspiration to add new features or custom buttons to give the app a new face.

Discussion:

- After completing the activity, discuss the concepts learned with the class, testing and projecting the code, if possible. Some questions that can guide the discussion:
 - What is a computer program? How does this program work?
 - Where in the code does the program read information from the internet?
 - Why is the value updated every 5 seconds?
 - What would happen if the piece of code that is inside the void loop function was passed to void setup?
 - What other applications could be created in order to create some practical solution using an application, internet of things and a microcontroller?

Credits:

Nathan Rabinovitch (LSITec/USP)