INTELLIGENT CONNECTED OBJECTS

SUBJECT	Connecting systems through Code IoT
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 work on the implementation of an application with an actuator
- To learn to connect electronic systems using the internet

NECESSARY MATERIALS Group 1

- Computer room with computers connected to the Internet
- NodeMCU board or similar
- Breadboard
- Jumpers
- Potentiometer

NECESSARY MATERIALS Group 2

- Computer room with computers connected to the Internet
- NodeMCU board or similar
- Breadboard
- Jumpers
- LEDs
- 220R resistor

INTELLIGENT CONNECTED OBJECTS

Conducting the activity:

In this activity, students will create projects that connect with each other from the Thingspeak platform. To conduct the activity, each project will be composed of 2 distinct groups: **Group 1** and **Group 2**.

Group 1 will be responsible for creating a circuit and code that will read the Thingspeak channel and **Group 2** will be responsible for creating a project that will read the values and create an application for the actuator.

- 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 card 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. Groups can be of the **Group 1** or **Group 2** type.
- 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.

Group 1

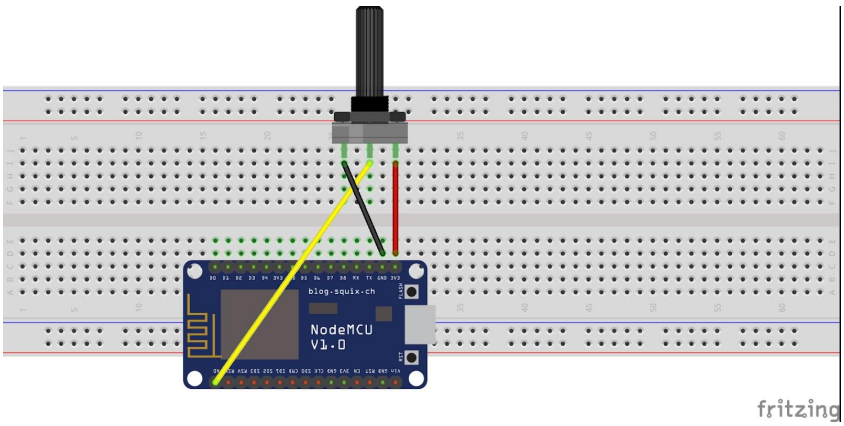
- Present the steps to transfer the code below onto the board.

Code:

```
// ThingSpeak and ESP8266 Libraries
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

// Your network data: name and password
char ssid[] = "xxxxxxxxx";          //  Network SSID Name
char pass[] = "xxxxxxxxx";   // Password

int status = WL_IDLE_STATUS;
WiFiClient  client;

int sensorValue; // variable that stores the value read by the sensor

// Information available on ThingSpeak: channel number and writing key
unsigned long canal = xxxx;
const char * WritingKeyAPI = "xxxxxxx";

void setup() {
  WiFi.begin(ssid, pass);
  ThingSpeak.begin(client);
}

void loop() {
  // Read input value on analog pin 0
  sensorValue = analogRead(A0);

  // Writes on ThingSpeak:
  // (channel number, field number, value, write key)
  ThingSpeak.writeField(channel, 1, sensorValue, WritingKeyAPI);

  delay(5000); // ThingSpeak accepts value updates every 5 seconds.
}
```

INTELLIGENT CONNECTED OBJECTS

- Check the circuit below:



- Once the network parameters, password, write and read keys, and Thingspeak channel have been changed, ask them to update the values read in the channel graphs on the Thingspeak platform.

Group 2

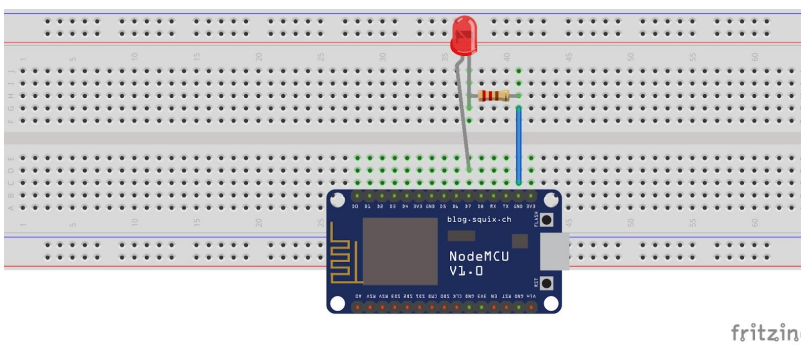
Present the steps to transfer the code below onto the board.

INTELLIGENT CONNECTED OBJECTS

Code

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

int reading;
int led = D7;

// network name and password
char 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(canal, 1, ChaveLeituraAPI);

  //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);
  }
}
```

- Check the project circuit below:



INTELLIGENT CONNECTED OBJECTS

- Once the network parameters, password, read/write keys and channel of Thingspeak have been changed, ask them to turn the LED on and off using the same system presented in lesson plan 1 of course 6.

  To connect the projects developed by the students, the channels and keys for writing and reading must be the same. Thus, **Group 1** will update the values from the potentiometer reading and **Group 2** will read this value and apply it to the LED.

Discussion:

- After completing the activity, discuss the concepts learned with the class, testing and showing 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 the communication between two microcontrollers and the internet of things?

Credits:

Nathan Rabinovitch (LSITec/USP)