Kreation.Arduino

On/Off Controller and Arduino LEDs

Introduction

Objective

Kreation.Arduino was created to help people build kreative projects using the Unity Game Engine and Arduino COMBINED and to make it easy to control Arduino from Unity.

Kreation.Arduino – OnOff Controller is a free, subset package with support for Arduino Uno and compatible boards only.

Please note that no guarantees can be made that this package will support every possible combination of hardware – there are too many Arduino compatible and similar boards out there. This has been tried with an Arduino UNO board.

Background

Arduino, and other micro-controllers compatible with the Arduino IDE, can be controlled by custom firmware, called sketches, or they can have the StandardFirmata sketch installed and then be controlled remotely, using the Firmata protocol.

Configuring Your Project

Setting up the Project for Serial Connections

The Arduino UNO has no firmware on it when you first buy it. You need to download the Arduino IDE to install software into the UNO board. In Arduino speak, the firmware source code is called a “Sketch.” It is a subset of the C++ programming language.

[Download Arduino IDE for Linux, Windows or Mac]

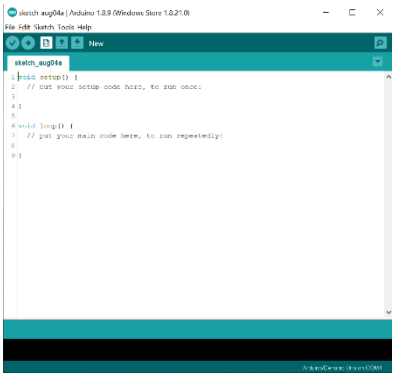
The software can be downloaded from the Arduino website, at this link:

https://www.arduino.cc/en/main/software

The Example sketches supplied with the IDE include Firmata support. Examples, libraries and support for other makes of boards can be installed or updated into the IDE. See the Arduino website for details on how to do this. For now, the IDE should be all we need to download.

Install StandardFirmata on UNO

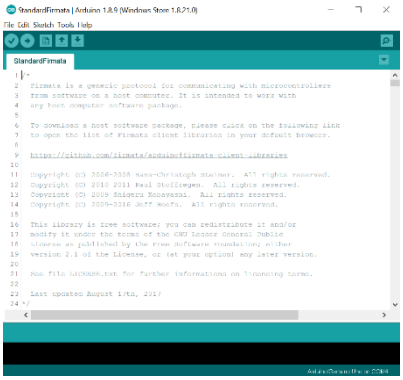
Run the Arduino IDE and you should see something like this screenshot (right).



We are not interested in writing a new Sketch. Instead we will use the StandardFirmata sketch that comes with the IDE.

To open the StandardFirmata sketch from the Examples:

1. Using the menu, click
   File | Examples | Firmata | StandardFirmata.
2. The IDE should open the StandardFirmata sketch in a new window. Also, you may have noticed there are other versions of the StandardFirmata sketch which support different types of connections. USB Serial is supported by the StandardFirmata sketch and that's all that is needed for the Uno.



3. Make sure the board is connected via USB and can be detected by the IDE. On Windows, the USB serial port will be **`COM4`** or **`COM5`**, may be different for other computers, and on Mac the port name will start with **`/dev/xxx`** where **`xxx`** could be a number of possibilities, depending on which USB serial driver and version of MacOS you have installed. In the menu, click **Tools** | **Port:** and try any listed port names.
4. Select the UNO board model by clicking on the **Tools** menu and finding the **Board:** item. If it says "**Board: Arduino/Genuino Uno**" then the right board is selected; otherwise hovering over the **Board** item will bring a sub-menu of boards – choose **Arduino/Genuino Uno**.
5. Now that we have the board and port configured compile and upload the StandardFirmata firmware into your UNO by clicking on the Upload (right arrow) button.

6. The black window at the bottom will show success or error messages for each step of compilation and upload.

This is all part of the normal Arduino operation so if something goes wrong, consult the Arduino website for problem solving and further information:

https://www.arduino.cc/en/Guide/HomePage

In Unity – The On/Off Controller

The Unity Asset is Installed

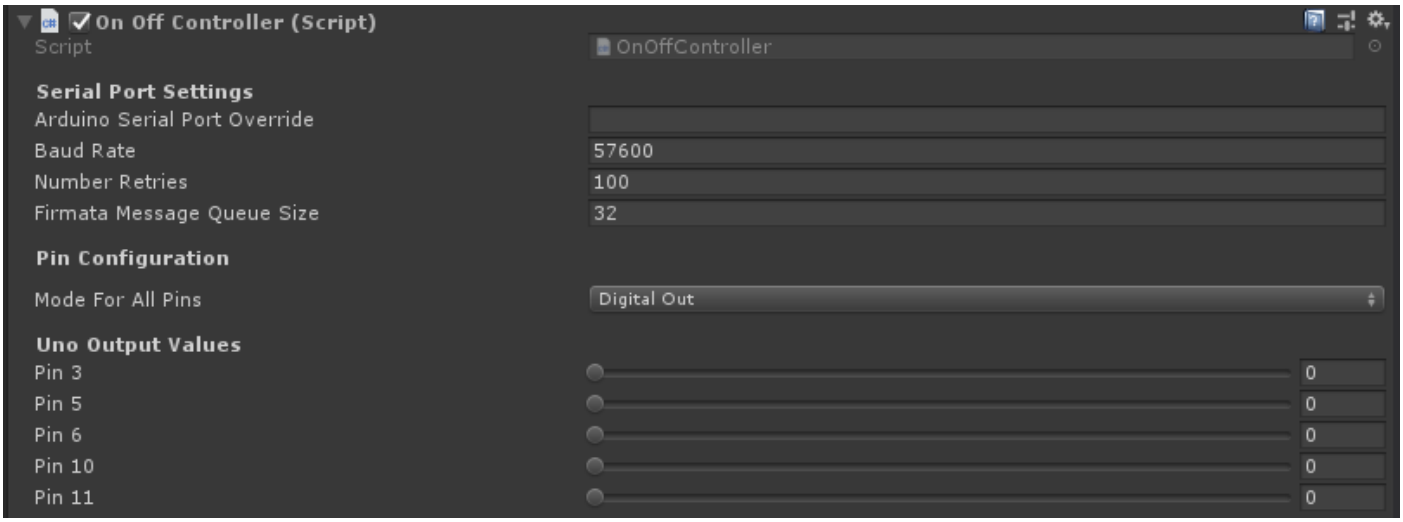
Assuming you installed this file and the Kreation.Arduino OnOffController through the Unity Asset Store – so no explanation needed on how that's done.

Starting a New Project

The On-Off Controller can be easily added to a Unity scene through the Unity menu **GameObject** | **Arduino** and selecting the **On-Off Controller** item. This will add a GameObject that is preconfigured with an OnOffController component.

The OnOffController only supports Arduino (or Genuino) UNO boards due to the pin out.

With your Arduino connected via USB serial, if you press play and go to Unity's inspector for the On Off Controller, you should see values for the UNO's pins 3, 5, 6, 10 and 11.



The controller will attempt to auto-detect your UNO but it can only do that if StandardFirmata is installed on the UNO and the USB serial driver is properly installed (in Windows this is usually already available). Also, you might connect multiple UNOs and this will pick the first it finds only.

Arduino Serial Port Override

The Arduino Serial Port Override setting allows you to enter the serial port name directly in case auto-detection isn't working – this might allow multiple UNOs to be connected or may help if auto-detection fails for some reason.

Baud Rate

The serial port speed - normally, you don't need to change this value. StandardFirmata assumes 57600 bits per second but you can change the number in the sketch to something faster but if you do, you should change this value to match.

Number Retries

When auto-detecting, the controller will try up-to this number of serial port reads looking for a Firmata response.

Firmata Message Queue Size

Normally, you can leave this unchanged.

The Kreation.Arduino Firmata code pre-allocates this many message buffers for queuing. Normally, it can keep up and re-use these buffers. You probably will not need to change this number and if it is not keeping up, you will see log messages appear in the Console window saying that it's getting low or ran out of buffers.

Mode For All Pins

At start-up, the OnOffController will use this one mode setting for all pins. The mode can be either analog out or digital out.

Digital will give an on signal of 5V or an off signal of 0V. "V" means volts.

Analog will vary the signal output voltage based on the pin value that is set. So, when LEDs are connected their brightness can be varied.

Pin X

The fields for pins 3, 5, 6, 10 and 11 are floating point numbers between 0 and 1.

For digital mode, 0.5 and greater causes 5V out and less than 0.5 will be off (0V).

For analog mode, the voltage will vary from 0V for a 0.0 value and 5V for a 1.0 value.

Note: these values cannot be changed through code but they can be animated using the Unity Animation window.

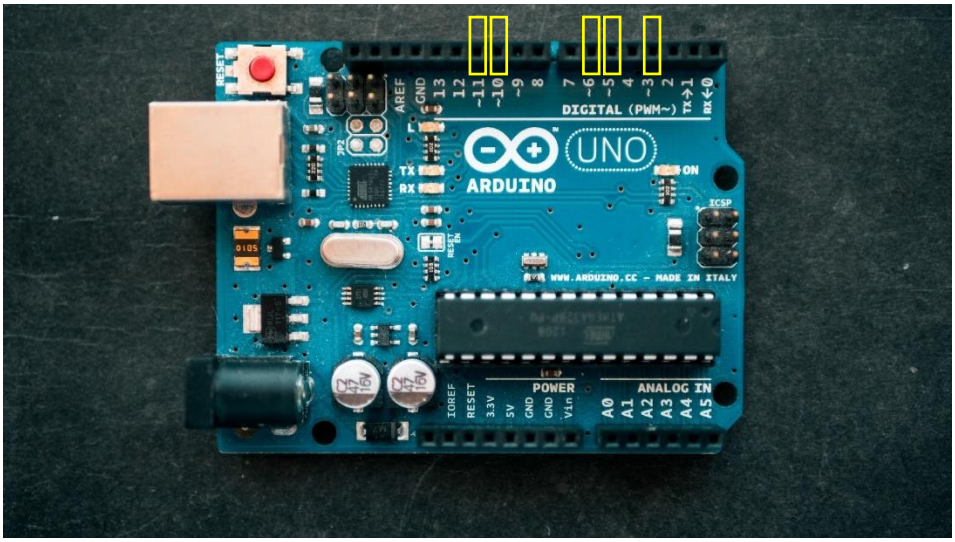


Photo by Harrison Broadbent on Unsplash

In Unity – The UnoBoard

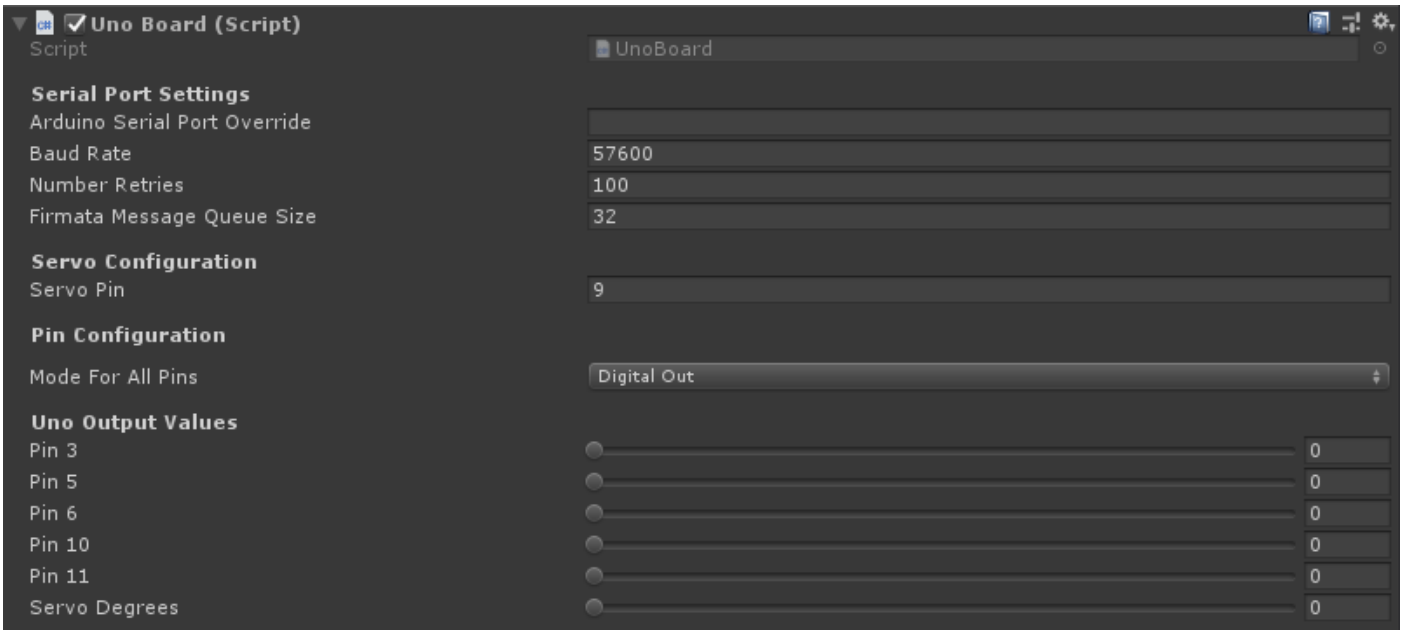
Description

The Arduino UNO GameObject can be created from the Unity menus – GameObject | Arduino | Uno Board. This creates an object with the UnoBoard component.

This is an extended version of the OnOff Controller with support for one servo.

Inspector Settings

The UnoBoard settings are the same as the OnOff Controller with two exceptions.



The Servo Configuration and the Servo Degrees output value.

Servo Pin

This setting allows you to choose a PWM pin as the control output for a servo motor.

Servo Degrees

The angle in whole degrees for the servo to be rotated. The angle must be a value from 0 to 180, inclusive.