Final Project for DVA480 Introduction to IoT Infrastructures

Student: Islombek Karimov

Project Description

An automated laser pointer toy for cats. The laser pointer moves in two dimensions to random positions with random wait times, to mimic an insect's behavior out in the wild. The toy is connected to the Arduino IoT Cloud and controlled (ON/OFF) via internet.

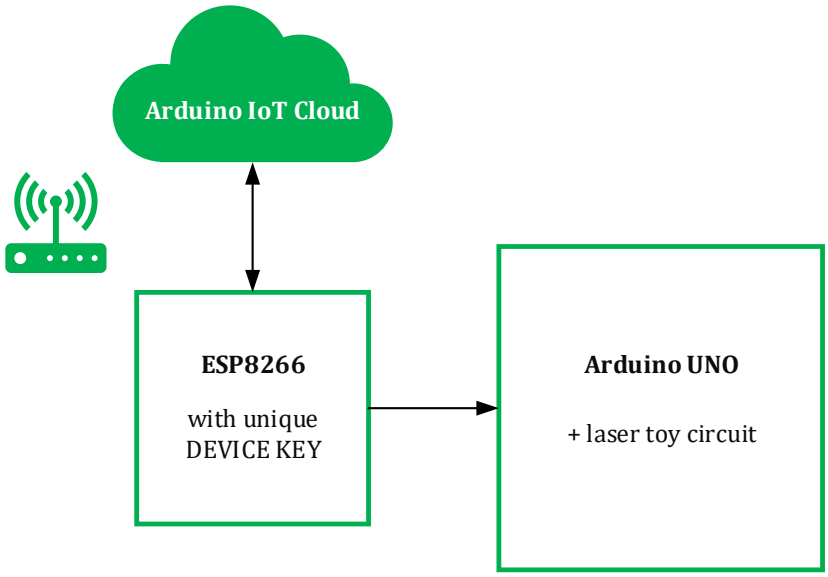
Demo video (with my cat that had become tired and didn't have energy to perform for the video) is found through this link: https://youtu.be/ZNs8fbQQlJg

All documentation (including this document) and code is uploaded to a GitHub repository and can be downloaded from there: https://github.com/idioteque11/WiFi_CatLaserToy

Design Description

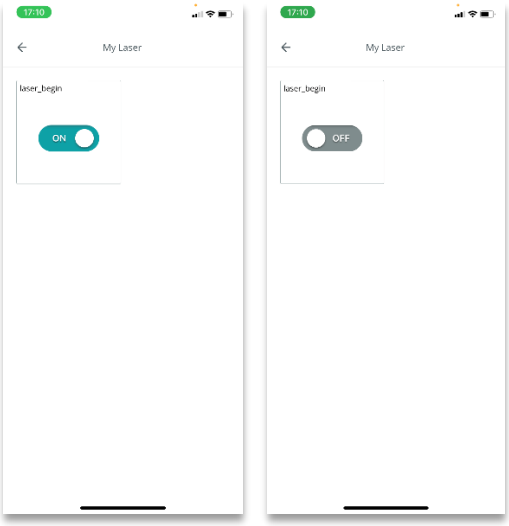
The laser toy module is based on the Arduino Uno microcontroller board. The servo motors along with the laser diode module are connected to the GPIO pins of the Arduino.

The Arduino communicates with an ESP8266 module that is connected to local home network via WiFi. This ESP8266 is set up in the Arduino IoT Cloud as a "Thing". During the setup of the ESP8266, the Cloud issues a unique Device Key that is associated with this specific ESP8266.



In the Arduino IoT Cloud Dashboard, a simple ON/OFF switch is created and is associated with a boolean (TRUE/FALSE) type variable in the ESP8266.

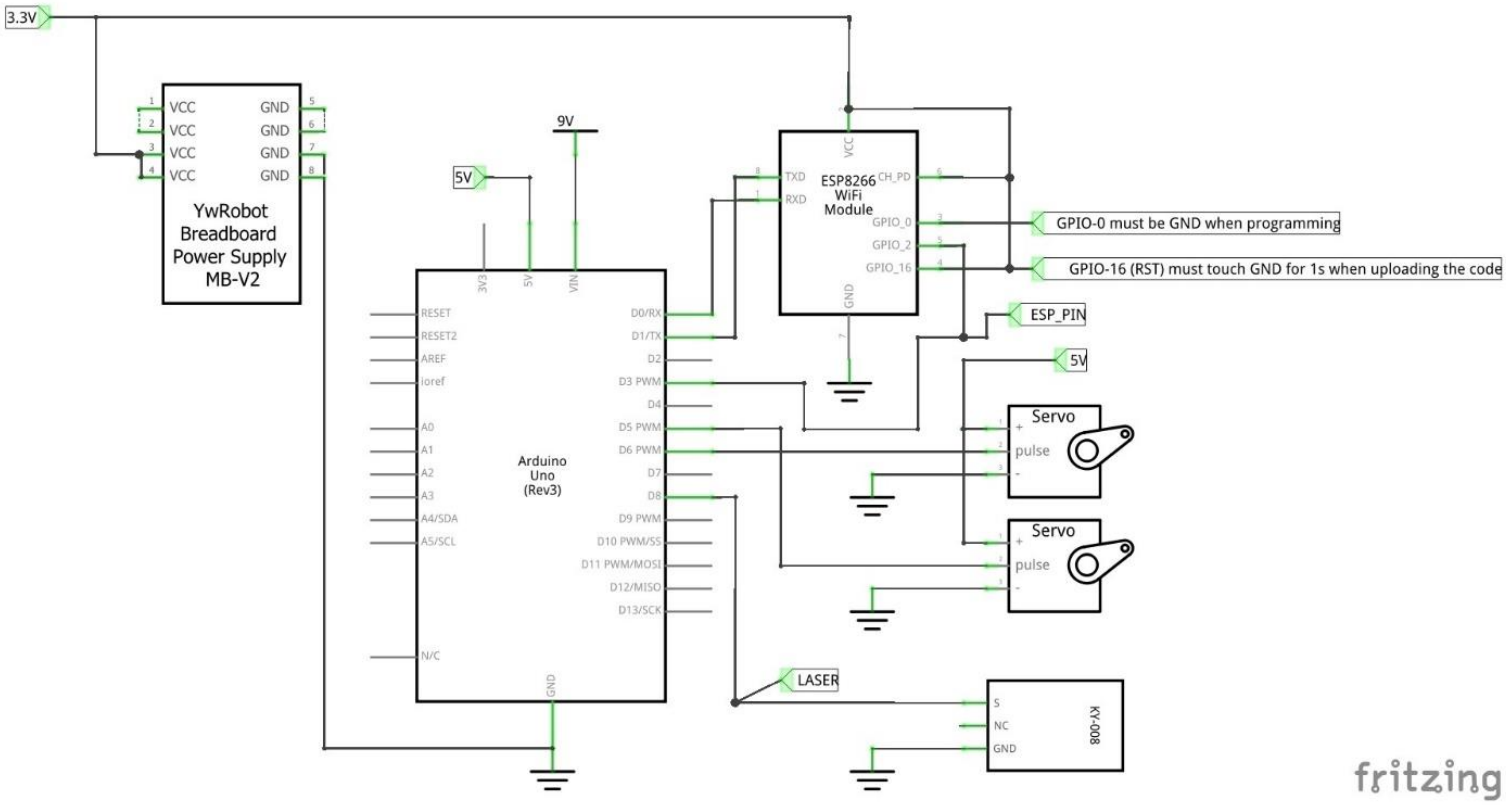
Thus, whenever this switch is toggled, the Cloud sends the data to the ESP8266 which then sends this signal to the Arduino. The Arduino processes this signal depending on user's instructions, in this case, to start and stop the motion of the servo motors to move the laser pointer to random positions.



The dashboard can be accessed either via Arduino IoT Cloud website https://create.arduino.cc/iot/things or via Arduino IoT Remote app on iOS or Android (see dashboard picture of the iOS app above).

Circuit Diagram

The Arduino is powered by a 9V battery.
The ESP8266 is powered by 3.3V bus coming from the Breadboard Power Supply Module.
The servo motors are powered by the 5V output pin of the Arduino.
All devices share a common 0V reference (GND) to reduce the crosstalk and other EMI related issues.



ESP8266 is connected to the system as follows:

ESP8266 pin	**Arduino pin**	**Comment**
TX	TX	This is not a uART configuration. Arduino only serves as a USB-to-serial interface for programming the ESP8266.
RX	RX	This is not a uART configuration. Arduino only serves as a USB-to-serial interface for programming the ESP8266.
CH_PD/EN	3.3V	For normal operation. 3.3V from the power supply.
RST	3.3V	For normal operation. 3.3V from the power supply.
VCC	3.3V	3.3V from the power supply, not the Arduino since Arduino's 3.3V doesn't output enough current that ESP8266 requires.
GPIO0	-	Not connected = for normal operation. GND = for programming.
GPIO2	3	Control pin to turn ON/OFF the motors. GPIO2 is set as output in ESP8266. Pin 3 is set as input in Arduino. See the codes for more info.

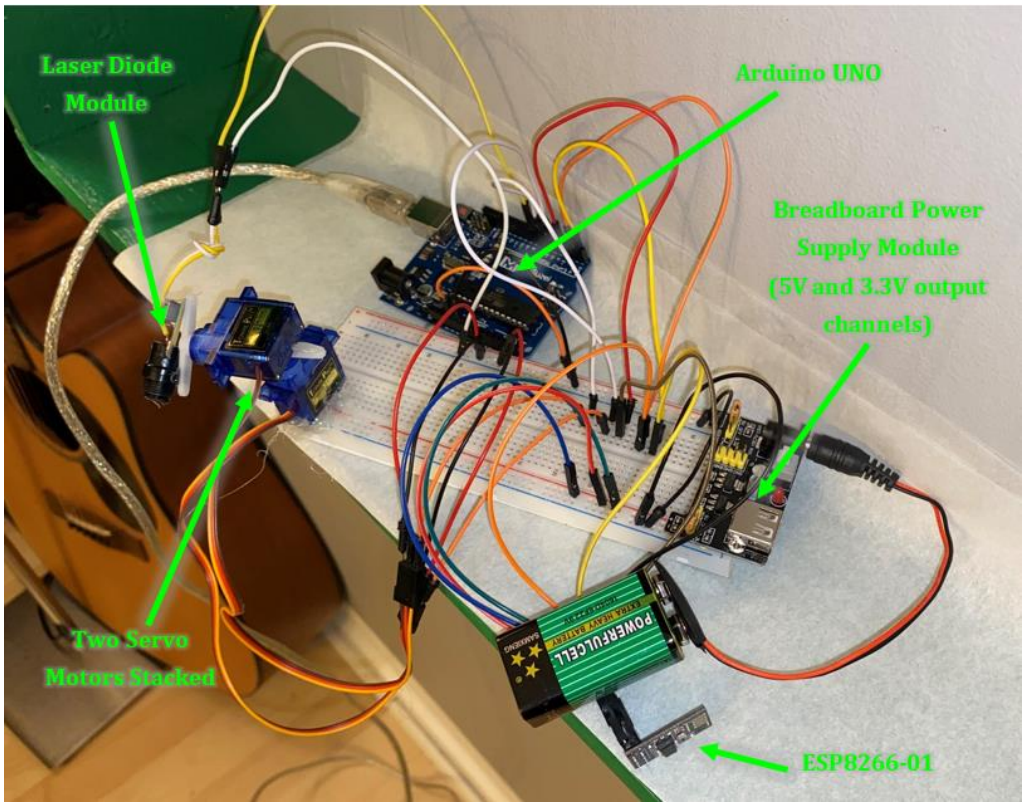
Servos and **the laser** are connected to the system as follows:

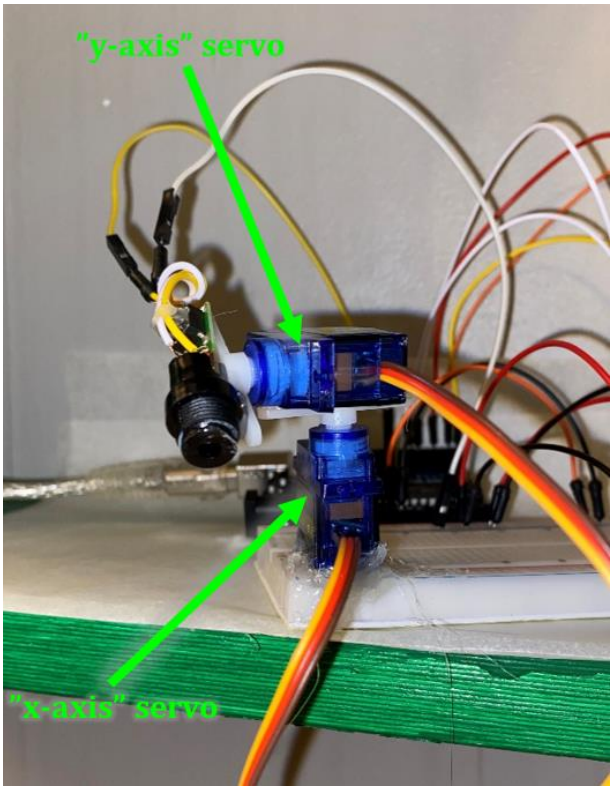
Servo X	**Arduino pin**	**Comment**
VCC	5V	The servo requires at least 5V for proper functioning.
GND	GND	Common ground for all devices.
PULSE	5	Control pin in the Arduino. Set as OUTPUT.

Servo Y	**Arduino pin**	**Comment**
VCC	5V	The servo requires at least 5V for proper functioning.
GND	GND	Common ground for all devices.
PULSE	6	Control pin in the Arduino. Set as OUTPUT.

Laser Diode Module	**Arduino pin**	**Comment**
VCC	8	Control pin in the Arduino. Set as OUTPUT. Note that there is already a current limiting resistor embedded in the Laser Diode Module. Thus, the power wire of the laser module can be directly connected to Arduino's IO pins.
GND	GND	Common ground for all devices.
NC	-	Not connected.

View of the System





Code

There are two devices that need to be programmed in this system: Arduino UNO and ESP8266 module.

The code is in the GitHub repository for this project:
https://github.com/idioteque11/WiFi_CatLaserToy

- The code for the Arduino is in *"LaserToy_WiFi"* folder.
    - There is also a calibration code for the Arduino: *"LaserToy_calibration".* It is used when calibrating the device to fit the "box" or the room in which it will be used. The "LaserToy_WiFi" code as it is now, is configured to my living room.

- The code for the ESP8266 is in *"esp8266_to_cloud"* folder.
    - Note that your ESP8266 device must first be configured in the Arduino IoT Cloud and its unique DEVICE KEY must be issued (serves as a unique address for the cloud server to locate your device and send data to/from it). Make sure to also type your WiFi network's SSID and password in the configuration header in the code, so that your ESP8266 can connect to your local network.

The Arduino can be programmed as it is generally done: via USB cable (since Arduino has a USB-to-serial interface) by uploading the code from the Arduino IDE or another IDE that supports Arduino libraries, such as VS Code.

The ESP8266 can be programmed with either a USB programmer device or by using Arduino as the USB-to-serial converter and uploading the code from the Arduino IDE.

Note: when programming the ESP8266 using Arduino, all pin connections to the Arduino are as during the normal operation (listed in the Circuit Diagram chapter above). However, GPIO0 pin must be connected to GND. Also, very important, a manual reset must be performed right before the code starts uploading:

> *The RST pin must be pulled low (touch the GND bus) for 1 sec right before the code starts uploading from the IDE, and pulled back high (set back to 3.3V).*

Once the code is successfully, uploaded, the GPIO0 pin must be disconnected from GND and manual reset (touch RST pin to GND for 1 sec and back to 3.3V) must be performed.

Revision History

Revision	Date	Author + comments
Rev 01	2021-12-07	Islombek Karimov: First issue of the document.

For any questions or suggestions, contact me at my email: ikarimovvv@gmail.com