

# API's library for Arduino board Simulator using Javascript

Presented by  
Karan Soren  
17MCMC12

Under Guidance of  
Dr.Nagendra Kumar Suryadevara,Associate Professor  
School of Computer and Information Sciences  
University of Hyderabad  
Hyderabad

22-07-2021



# Contents

- ▶ Introduction
- ▶ Arduino
- ▶ Arduino's inbuilt Components
- ▶ Components of the Arduino
- ▶ Johnny-Five's API
- ▶ Simulator - Proteus Design Suite
- ▶ API Method
- ▶ Schematic Design
- ▶ Task Analysis
- ▶ Execution and Results
- ▶ Demo
- ▶ Conclusion and Future Works



# Aim and Objectives

## ► Aim

- ▶ This project is intended to teach students or any users how to use the Johnny-Five Library.
- ▶ And how to link the provided Johnny-Five's API library with the Arduino and its components using a simulator.
- ▶ Students can access Arduino and other devices without touching them using the simulator.

## ► Objectives

- ▶ Create schematics with Proteus Design Suite.
- ▶ Using modules - Johnny-Five.
- ▶ Connecting the Components using Simulator.
- ▶ Implementing the Johnny-Five API via Node js



# Introduction

- ▶ Exploring a working Arduino system using the API library.
- ▶ Students had to think about buying an Arduino Board.
- ▶ Now, there is a simulator.
- ▶ Using the Johnny-Five, control the Arduino with components such as LEDs, DC-Fan, Servo Motor, and LCD.
- ▶ After using the Johnny-Five library, students will have a basic knowledge of the Arduino Board Simulator and its features as it is free source code.
- ▶ They will have the opportunity to realize imaginative ideas.



# Arduino

- ▶ A microcontroller-based open-source electronic prototyping board.
- ▶ A physical programmable circuit board.
- ▶ Powered from either a 9V battery or a power supply, via a USB connection to the computer.

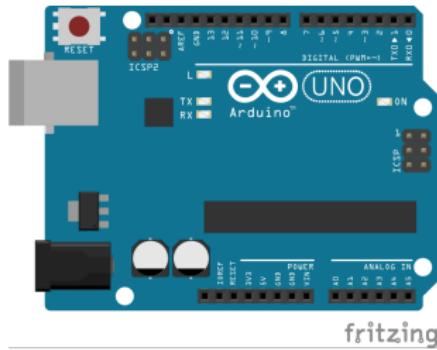


Figure 1: Arduino UNO



# Arduino inbuilt Components

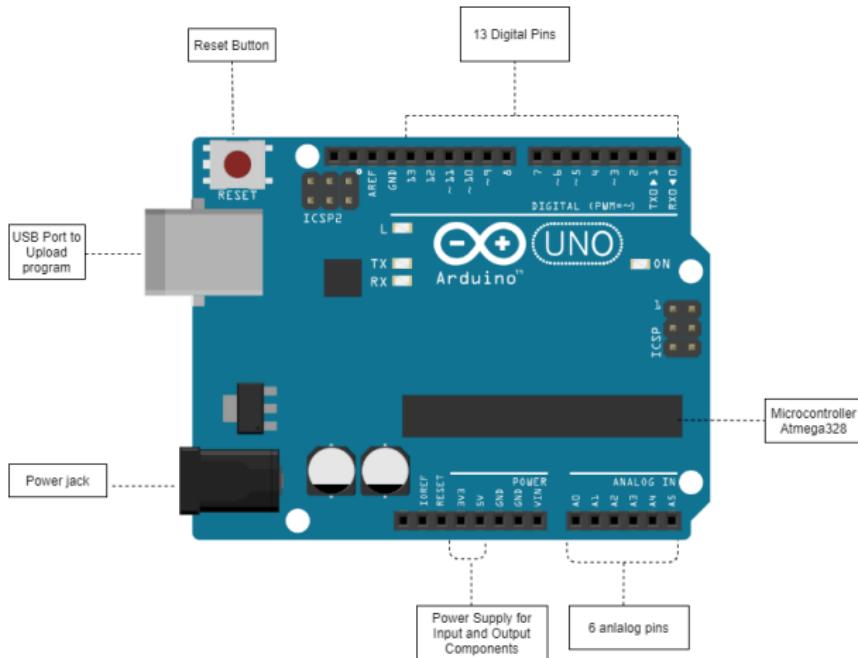


Figure 2: Components of Arduino



# Components of the Arduino

- ▶ General Component
  - ▶ Resistors
- ▶ Output Components
  - ▶ LEDs
  - ▶ Fan Motor
  - ▶ Servo Motor
  - ▶ LCD
- ▶ Input Components
  - ▶ TH02
  - ▶ Push Button
  - ▶ Matrix Keypad



# Johnny-Five's API

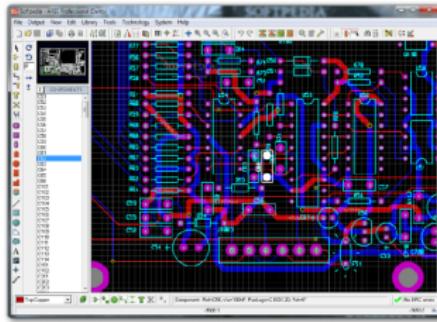
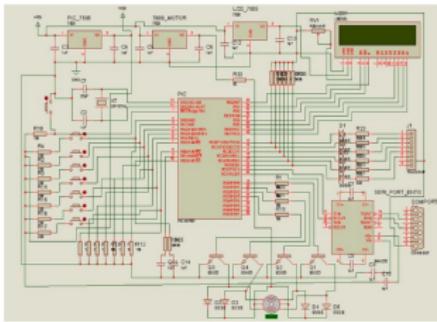


- ▶ Javascript robotics and IoT platform.
- ▶ Allows us to communicate with microcontrollers using node js.
- ▶ Maintained in a GitHub repository
- ▶ Work well with popular application libraries such as express js and socket io.



# Simulator - Proteus Design Suite

- ▶ Simulation software for Arduino.
- ▶ For the schematic design
- ▶ Manufacturing PCB (Printed Circuit Board) designs.

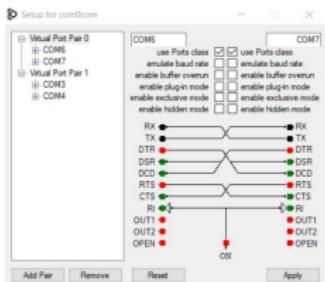


# API Method

- ▶ com0com
- ▶ Arduino Standard Firmata
- ▶ Node js
- ▶ Visual Studio



# com0com

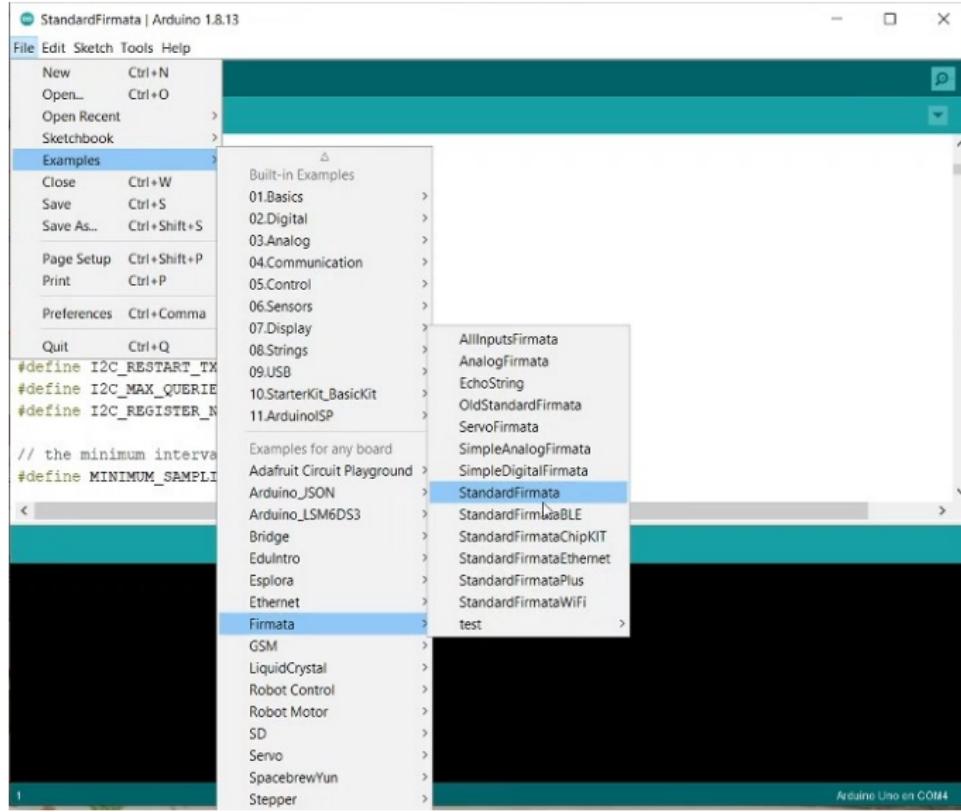


```
C:\Users\Karan Soren>mode COM1: Baud=57600 parity=n data=8 stop=1
status for device COM1:
-----
Baud:      57600
Parity:    None
Data Bits:  8
Stop Bits: 1
Flow Control: XON/XOFF: OFF
              CTS handshaking: OFF
              DSR handshaking: OFF
              DSR sensitivity: OFF
              DTR circuit:   ON
              RTS circuit:  ON
```

```
C:\Users\Karan Soren>mode COM1: Baud=57600 parity=n data=8 stop=1
status for device COM1:
-----
Baud:      57600
Parity:    None
Data Bits:  8
Stop Bits: 1
Flow Control: XON/XOFF: OFF
              CTS handshaking: OFF
              DSR handshaking: OFF
              DSR sensitivity: OFF
              DTR circuit:   OFF
              RTS circuit:  ON
```



# Arduino Standard Firmata



# Node Js

Latest LTS Version: 14.17.3 (includes npm 6.14.13)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

| LTS<br>Recommended For Most Users  | Current<br>Latest Features   | Source Code  |
|--|--|--|
| <br>Windows Installer | <br>macOS Installer |  |
| node-v14.17.3-x64.msi  | node-v14.17.3.pkg  | node-v14.17.3.tar.gz   |
| Windows Installer (.msi)   | 32-bit   | 64-bit   |
| Windows Binary (.zip)  | 32-bit   | 64-bit   |
| macOS Installer (.pkg)   | 64-bit   |  |
| macOS Binary (.tar.gz)   | 64-bit   |  |
| Linux Binaries (x64)   | 64-bit   |  |
| Linux Binaries (ARM)   | ARMv7  | ARMv8  |
| Source Code  | node-v14.17.3.tar.gz   |  |



# Visual Studio Code

The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is "server.js - Visual Studio Code". The left sidebar has icons for file operations like Open, Save, Find, and Refresh. The main editor area displays the following code:

```
37     console.log("Message received msg:"+msg);
38   });
39   // led toggle function
40   socket.on("led-toggle", function(state){
41     //jhonny-five code here
42     if(state)
43       myLed.on();
44     else myLed.off();
45     console.log("LED "+state);
46     myPin.query(function(state){console.log("Pin 13 state:"+state.state); sc
47   });
48 });
49 });
50
```

Below the editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing a PowerShell session with the command `npm install johnny-five`. The output of the command is displayed:

```
PS C:\Users\Karan Soren\Documents\UoH\Sem_6\Project\single_led_control> npm install johnny-five

removed 71 packages, changed 1 package, and audited 92 packages in 5s

found 0 vulnerabilities
PS C:\Users\Karan Soren\Documents\UoH\Sem_6\Project\single_led_control>
```

The bottom status bar shows the file path "C:\Users\Karan Soren\Documents\UoH\Sem\_6\Project\single\_led\_control\server.js", line 1, column 1, and other details like "Spaces: 2", "UTF-8", "LF", "JavaScript", and icons for file operations.



# Schematic Design

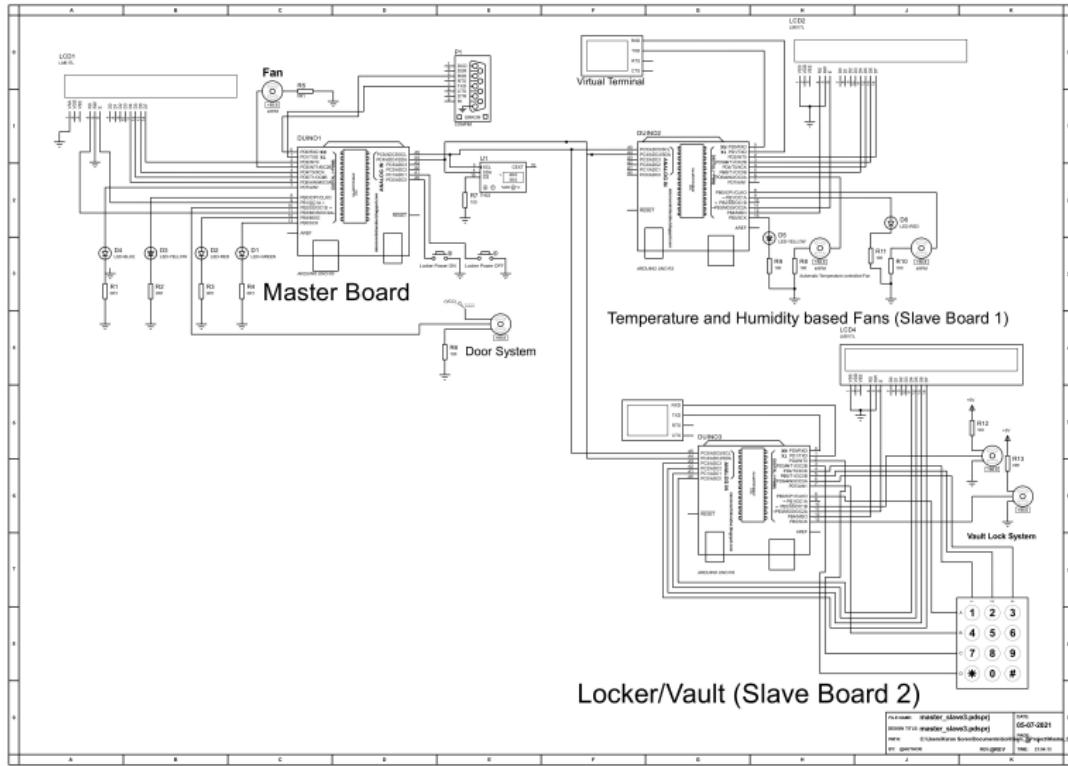


Figure 3: Schematic Design using Proteus Design Suite

Presented by Karan Soren 17MCMC12

API's library for Arduino board Simulator using Javascript

# Task Analysis

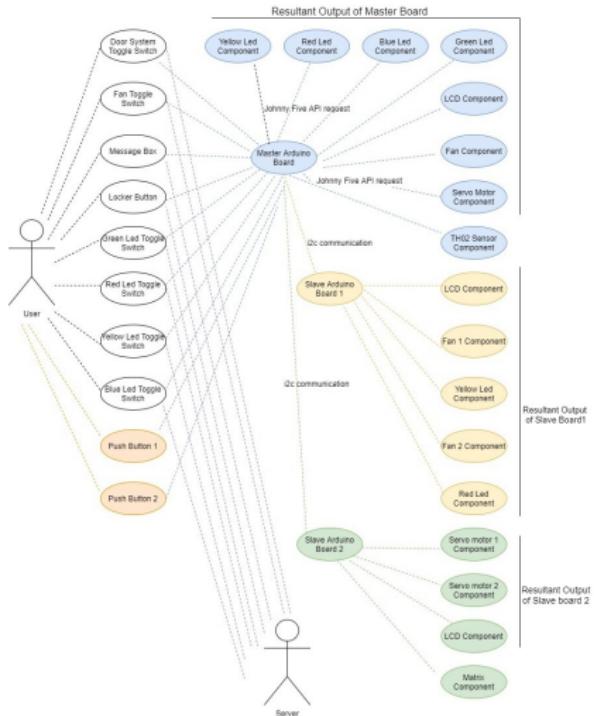
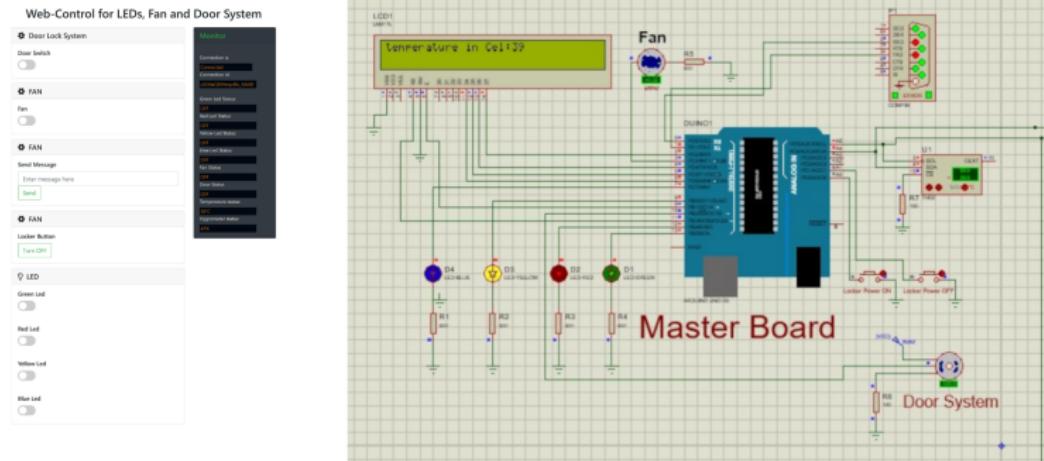


Figure 4: Use-Case Diagram

## Execution and Results

- ▶ Press the play button in Proteus Design Suite.
  - ▶ Enter node javascript\_filename.js in the terminal.
  - ▶ Open the browser and enter it: <http://localhost:8080>.



# Demo



Demo



# Conclusion and Future Work

## ► Conclusion

- ▶ Best for teaching beginners to work with Arduino using Simulator.
- ▶ Learning all those components and the API library that uses it.
- ▶ learn important of APIs on Arduino.
- ▶ The johnny five library is really useful for writing Arduino code.

## ► Future Work

- ▶ There is no need to buy an Arduino board and its equipment.
- ▶ This can be a blueprint for the embedded software engineer.
- ▶ Perhaps I will be one of the students who will later build real projects,



# Thanks

