# REMOTELY CONTROLLED

# SELF-BALANCING

# MINI BIKE

**Team 22**

**Member: Jiaming Xu(jx30), Will Chen(hongyuc5), Eric Tang(leweit2)**

**Spring 2024**

**TA: Jason Zhang**

**ECE445**

# Introduction

## 1.1 Problem:

Bike Share and scooter share have become more popular all over the world these years. This mode of travel is gradually gaining recognition and support. Champaign also has a company that provides this service called Veo. Short distance traveling with shared bikes between school buildings and bus stops is convenient. However, since they will be randomly parked around the entire city when we need to use them, we often need to look for where the bike is parked and walk to the bike's location. Some of the potential solutions are not ideal, for example: collecting and redistributing all of the bikes once in a while is going to be costly and inefficient; using enough bikes to saturate the region is also very cost inefficient.

## 1.2 Solution:

We think the best way to solve the above problem is to create a self-balancing and moving bike, which users can call bikes to self-drive to their location. To make this solution possible we first need to design a bike that can self-balance. After that, we will add a remote-control feature to control the bike movement. Considering the possibilities for demonstration are complicated for a real bike, we will design a scaled-down mini bicycle to apply our self-balancing and remote-control functions.

## 1.3 Visual Aid:

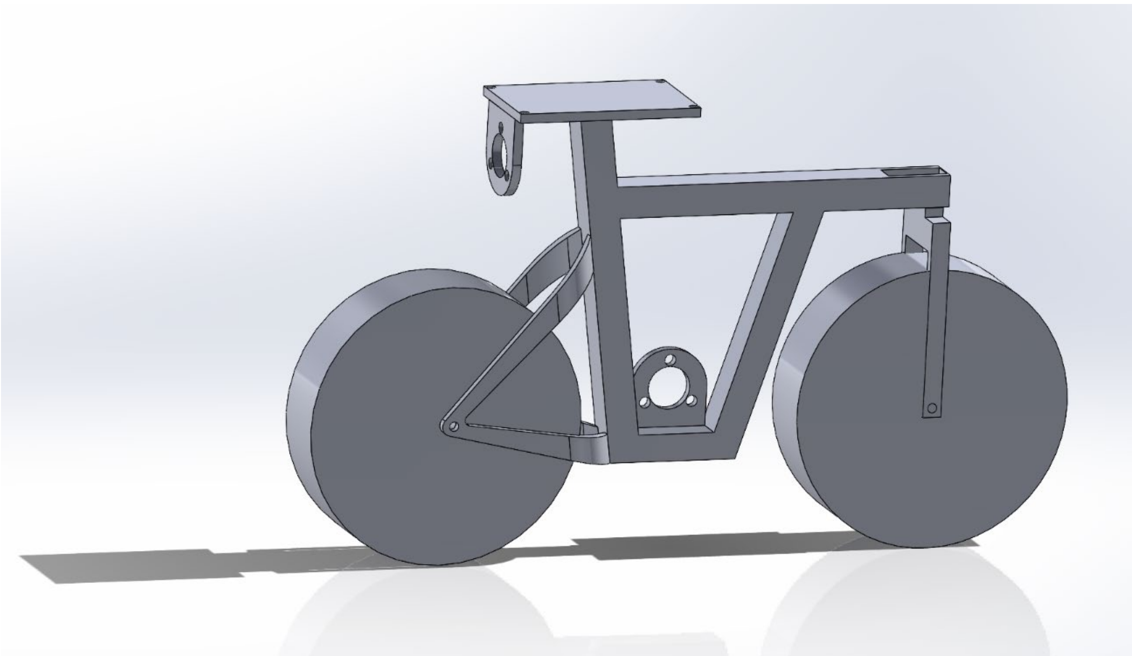

Figure 1: Visual Aid

## 1.4 Three High-level requirements list:

a. The bicycle should be able to remain self-balanced when the power is on.
b. The bicycle should be able to self-regulate lateral force disturbances within +- 10 degrees and continue to maintain balance.

c. The bicycle should be remotely controlled by the controller within a radius of 5 m from a bicycle and can maintain self-balancing during controlled movement.

# Design

## 2.1 Block Diagram:

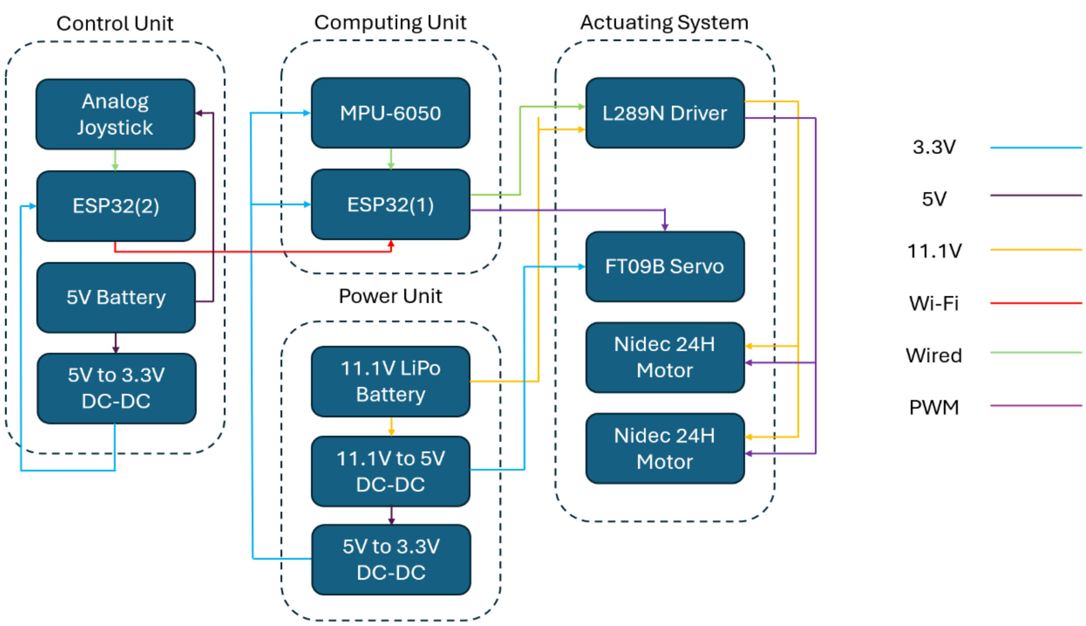


Figure 2: Block Diagram

## 2.2 Subsystem Description

### 2.2.1 Power Unit

Overview:

We plan to use one 11.1V LiPo Battery to power all 3 motors and one 11.1V to 3.3V DC-DC converter to power all other electronics, such as microcontroller ESP32-S3-WROOM[2], and MPU-6050.

Requirement:

d. The 11.1V LiPo Battery is supposed to power all 3 motors.
e. The 11.1V to 3.3V DC-DC converter is supposed to provide stable 3.3V+-0.3V to microchips and IMU.

### 2.2.2. Control Unit

Overview:

We plan to design a Wi-Fi 2.4GHz remote controller using two ESP32-S3-WROOM microcontrollers and a two-axis pushbutton rocker sensor module by joystick potentiometer to implement the function of remote control. One ESP32-S3-WROOM serves as the transmitter which functions as the primary input source, providing control inputs such as acceleration, braking, and steering. It communicates these inputs wirelessly to another ESP32-S3-WROOM microcontroller function as a receiver. The ESP32-S3-WROOM Microcontroller decodes and processes these data packets to extract control commands, including steering angle, brake intensity, and any additional functions mapped to the controller buttons. The communication between two ESP32 microcontrollers will be coded based on ESP_NOW, which is a wireless peer-to-peer Wi-Fi communication protocol. ESP-NOW utilizes Wi-Fi technology, which typically offers better range and stability compared to Bluetooth or traditional RF protocols. It also supports higher data rates compared to Bluetooth and RF protocols, allowing for faster transmission of control commands and data between the remote controller and the minibike, which is beneficial for our project since it requires real-time control.

Requirements:

1. The bicycle can be operated by the remote controller within a radius of 50 m of a bicycle.
2. The remote controller must maintain stable communication with the bicycle with a delay $\leq$ 20ms.

**2.2.3. Computing Unit**

Overview:

We plan to use an MPU6050 6-axis Gyro and accelerometer sensor to obtain the necessary information about the bike to calculate its tilted angle. The MPU6050 will take 3.3V from the 11.1V-3V converter as input, and it will also be connected to the microcontroller ESP32-S3-WROOM [2]. The microcontroller ESP32-S3-WROOM [2] will decode control information received from the controller, compute the bike's tilted angle with information from the MPU6050, and output the corresponding control signals to each motor.

Requirement:

1. The microcontroller must compute the correct tilted angle of the bike with information from the MPU unit with a tolerance of 1 degree despite external disturbance.
2. The MPU6050 must detect the correct orientation and acceleration of the bike.

**2.2.4. Actuating Unit**

Overview:

Currently, we plan to use L298N as the motor driver. The controlling signal from the microprocessor ESP32-S3-WROOM[2] will be sent to motor drivers and then passed to each motor. All 3 motors will take 11.1Volts as input voltage. We will use FT90B Servo[3] from FeeTech to control the steering of the bike. One Nidec 24H BLDC Servo motor[4] will be connected to the customized reaction wheel to achieve self-balancing, and the other Nidec 24H BLDC servo motor will be connected to the rear wheel of the bike to achieve the movement of the bike.

Requirements:

1. FT90B Servo [3] must provide correct steering based on the controller inputs.
2. The reaction wheel and connected BLDC servo motor should provide self-balancing to the bike.
3. The second BLDC servo motor should provide movement to the bike based on the controller inputs.

11V motors connected parallel to one 11V power source. A voltage regulator(LM2596)[5] converts to 5 V, which powers all control units and steering servo motors.

# Tolerance Analysis:

## 3.1 Parts Current

| Part | Worst Case Current Draw @ 5 V | Total Current:<br>1.005A |
|---|---|---|
| Processor | 250mA | |
| Motor Controller | 50mA | To be safe we give an extra 20% of the designed current needed. Thus, we will design with 1.2A |
| Servo Motor | 700mA | |
| accelerometer | 5mA | |

## 3.2 Variables

| **Variable** | **Value** |
|---|---|
| Maximum temperature(LM2596) | 125 C |
| Efficiency(eff) when operating @ 5V | 85% |

| | |
|---|---|
| Voltage output($V_{out}$) | 5V |
| Junction-to-case resistance($\Theta_{jc}$) | 2 C/W |
| Thermal resistance($\Theta_{ca}$) | 30 C/W |
| Environment temperature($T_{env}$) | 38 |

### 3.3 Calculation of heat tolerance

First we calculate the power output of the system:

$$P_{out} = V_{out} * I_{in} = 5V * 1.2A = 6W$$

Then we use the efficiency of voltage regulator to calculate the power dissipated, and assume all power dissipated into the generation of heat

$$P_{diss} = P_{in} - P_{out} = P_{out}/eff - P_{out} = 1.06W$$

Now we can use the power that generates heat and the environment temperature to calculate a theoretical value for operating temperature:

$$T_{op} = P_{diss} * (\Theta_{jc} + \Theta_{ca}) + T_{env} = 1.06 *(2 + 30) + 38 = 71.92\ C < 125\ C(T_{max})$$

The operating temperature is well below the maximum temperature of LM2596. Thus, we can conclude that our system will operate under the limit of LM2596, we have around 70% tolerance of the original design.

# Ethics and Safety:

Throughout this project, we adhere to the IEEE Code of Ethics. We will "uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities"[1]. We will "treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others"[1]. We will "strive to ensure this code is upheld by colleagues and co-workers"[1].

We recognize that there are some potential safety concerns during this project. The workstation must be kept clean and organized, and tools such as soldering irons must be shut down immediately after use. We will clean up the workstation right after use take a picture upload it to Google Drive and annotate the date.

When handling motors, we must ensure that we have a clean, spacious, space area. We need to check the number of people around, potential electrical dangers such as ponds or wires, and the cleanliness of the surface the cart is going to be running. Safety is always the top priority throughout this project, and every team member needs to follow the procedure above.

The Lithium battery must be kept safely inside a fire-retardant charging bag for storage and charging. For storage a few days long, we need to find a dry and cool place that won't be directly hit by sunlight. For storage longer than a few days, we need to follow all the instructions above and use a meter to measure the voltage and make sure their voltage is around 4.4V. When charging the battery, we can only use the provided charger from the manufacturer. During usage, we will charge the 11V battery frequently to ensure the power of our 11V motors.[2]We will do voltage testing every time after charging the battery and document the voltage, and visual inspection every time before and after charging the battery. When batteries are not in use we need to disconnect them from the cart and store it in a dry enclosed transparent box. If the battery shows any signs of leakage or malfunction, we must immediately stop using it and depose it appropriately.

**Reference:**

[1] "IEEE Code of Ethics." *IEEE*, https://www.ieee.org/about/corporate/governance/p7-8.html

[2] "ESP32-S3-WROOM-1" datasheet, Version1.3 EspressifSystems https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[3] "FT90B" FreeTech https://www.parallax.com/product/3v-digital-micro-servo-standard-ft90b/

[4] "Nidec CMC_24" datasheet https://www.nidec.com/en/product/search/category/B101/M102/S100/NCJ-24H-24-01/

[5] LM2596, Texas Instrument https://www.ti.com/lit/ds/symlink/lm2596.pdf