



Boston University
Electrical & Computer Engineering
EC464 Senior Design Project

Final Prototype Testing Plan

OptiSync

by

Team 9

OptiSync

Team Members

Bailey Brake bbrake01@bu.edu

Ruohui Huang rhhuang@bu.edu

Emily Lampat lampat26@bu.edu

Khang Le khang@bu.edu

Yuri Zhang zyuri@bu.edu

1.0 Required Materials

Hardware:

- T0 model
- T9 model
- Oscilloscope
- Cables
 - BNC to hook
 - USB-A to mini USB
 - USB-C to micro-USB

Software:

- Arduino IDE
 - Arduino script to generate 40Hz light frequency.
 - Take buttons as input to write into LED output, acting as power source.
 - T0 model: ON/OFF
 - T9 model: brightness adjustability

2.0 Set Up

2.1 T0 model set up

Our T0 model consists of an arduino nano microcontroller that receives the button on/off input and outputs a flickering rate of 40Hz for our LED. There is a MOSFET that will boost the current supplied to the LED without surpassing the microcontroller output capabilities.

2.2 T9 model set up

Our T9 model consists of an adafruit trinket microcontroller that receives increase/decrease input from two buttons, and outputs a flickering rate of 40Hz for our LED. There is also a MOSFET in this circuit that boosts the current supplied to the LED.

3.0 Pre-Testing Setup Procedure

3.1 T0 model pre-testing setup

Pre-testing setup procedure is fairly simple. Please refer to the Figures 1 and 2 to ensure that the setup is correct. All that is needed to do is connect the microcontroller into a wall plug and place the phone in the holder attached to the light panel.



Figure 1. Phone placed in holder for T0 model LED Panel

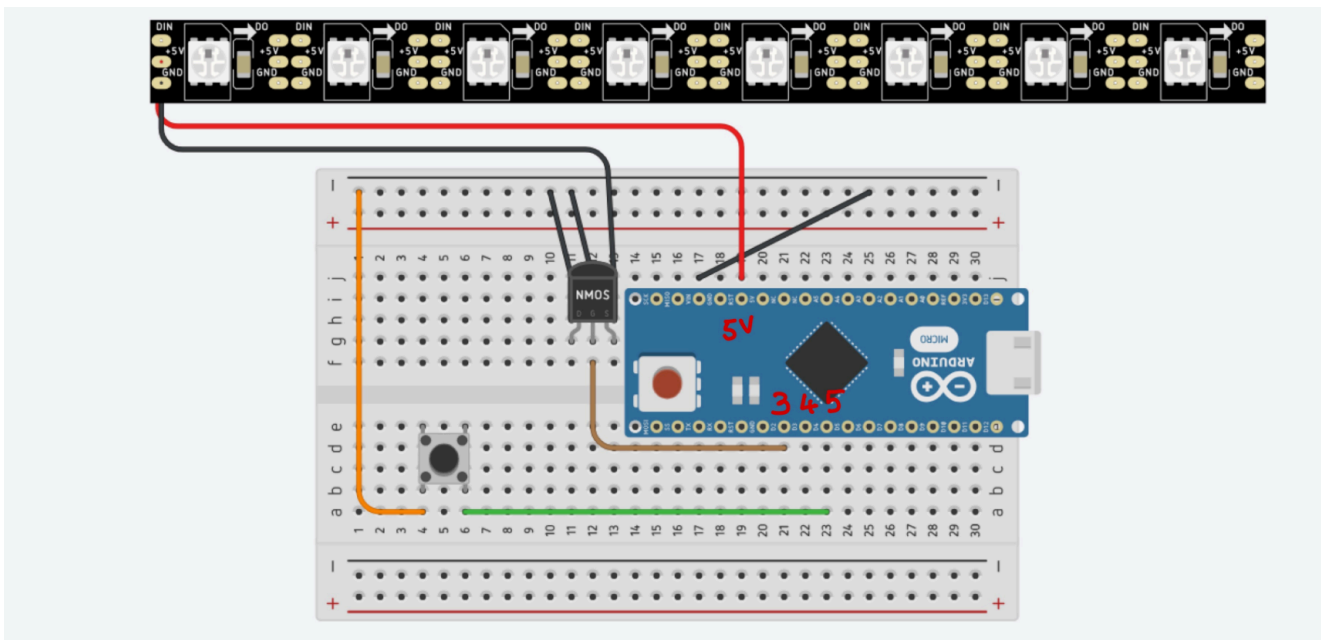


Figure 2. Circuit Diagram of the T0 Model

3.2 T9 model pre-testing setup

This is the hand-held option that we have made, so setup is a little bit different from the T0 model. Please refer to Figure 3 to ensure that the setup is correct. Attach the T9 model to the back of the phone using velcro and plug the microcontroller into the power port of the phone. The cable would have to be micro-usb to usb-c connection.

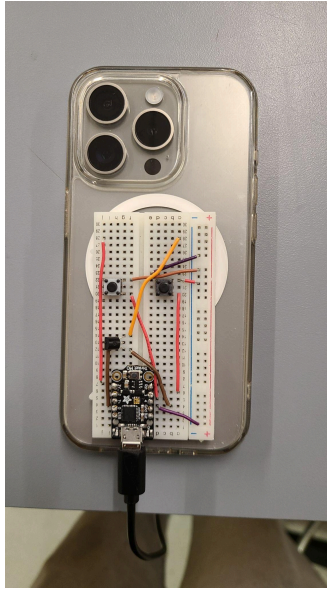


Figure 3. T9 model attached to the back of the phone.

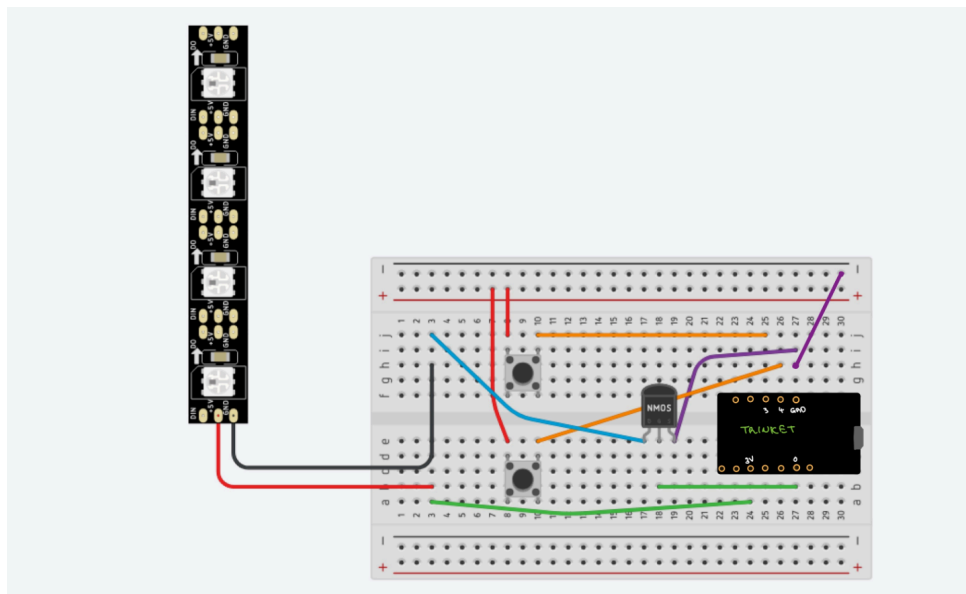


Figure 4. Circuit Diagram of the T9 model

4.0 Testing Procedure

4.1 T0 model:

1. Ensure Arduino is connected to a wall outlet power source.
2. Verify that the LED status on the Arduino is lit up.
3. Press the button multiple times to verify the LED is receiving an ON/OFF signal.
4. Ensure that the phone holder is able to support the phone by placing the phone onto the holder attached to the panel.
5. Use probe wires to verify frequency with oscilloscope

4.2 T9 model:

1. Attach T9 model to the back of a smartphone
2. Ensure Arduino is connected to the phone power port.
3. Verify that the LED status on the Arduino is lit up.
4. Press the increase button to verify the LED is receiving the “increase brightness” signal.
5. Press the decrease button to verify the LED is receiving the “decrease brightness” signal.
6. Ensure the LED is on
7. Use probe wires to verify frequency with oscilloscope

5.0 Measurable Criteria

The criteria for having a successful running prototype are as follows

5.2 T0 model:

- I. The circuit board should be intact with the LED panel.
- II. The LED should be at the front of the panel connected to the circuit board.
- III. Upon plugging in the microcontroller and pressing down the button, the LED should start flashing.
- IV. When toggling with the button, the LED should turn on and off within <1 second.

5.2 T9 model:

- I. The circuit board should be intact with the LED panel.
- II. The LED should be facing the user when using the phone
- III. Upon plugging in the microcontroller and pressing the “Increase Brightness” button, the light should turn on and get brighter until it reaches maximum brightness.
- IV. When pressing the “Decrease Brightness” button, the LED brightness should dim until it is turned off.

6.0 Hardware Pinout

T0 model:

Pin Number	Usage/Description
3	White LED Output
4	LED ON/OFF Button Input

T9 model:

Pin Number	Usage/Description
0	White LED Output
4	Increase Brightness
3	Decrease Brightness