CS/EE 120B Custom Laboratory Project Report

Audio Decibel Safety Mechanism

Krishaan Patel

03/17/22

Introduction

This project is an audio safety mechanism that measures the audio level currently playing in an environment and notifies the user if it is at an unsafe listening level. The system will show the user the current audio level of their environment, and can notify the user of an unsafe listening level through a light, sound, and text of the intensity level.

Complexities

The first complexity is the uxcell Sound Microphone Sensor which tracks incoming audio levels as input. The second complexity is the passive buzzer which could be toggled via a button to alert a sound whenever an unsafe audio level is playing.

The final complexity is the Nokia 5110 LCD which displays the current audio level received from the mic input as well as an interface showing the intensity level for if the audio is safe or not.

User Guide

A user would operate with this system by placing the microphone of the system in front of some form of a speaker (such as a physical one or one from a device). The microphone will then take in the currently playing audio level as input and will give multiple forms of output which change depending if the audio is playing at a safe listening level. The first output the user can see is the LED lighting up when unsafe, so the user knows they must decrease their audio level. The second output the user can see is text visualized on the Nokia 5110 LCD which will give them a specific audio level value and an Intensity reading of "LOW" or "HIGH" for when the audio is safe or not. Finally, the user can press the button to toggle the passive buzzer to notify the user via a buzzing sound for when the audio reaches an unsafe listening level. If the button is pressed again it will toggle the buzzer back off.

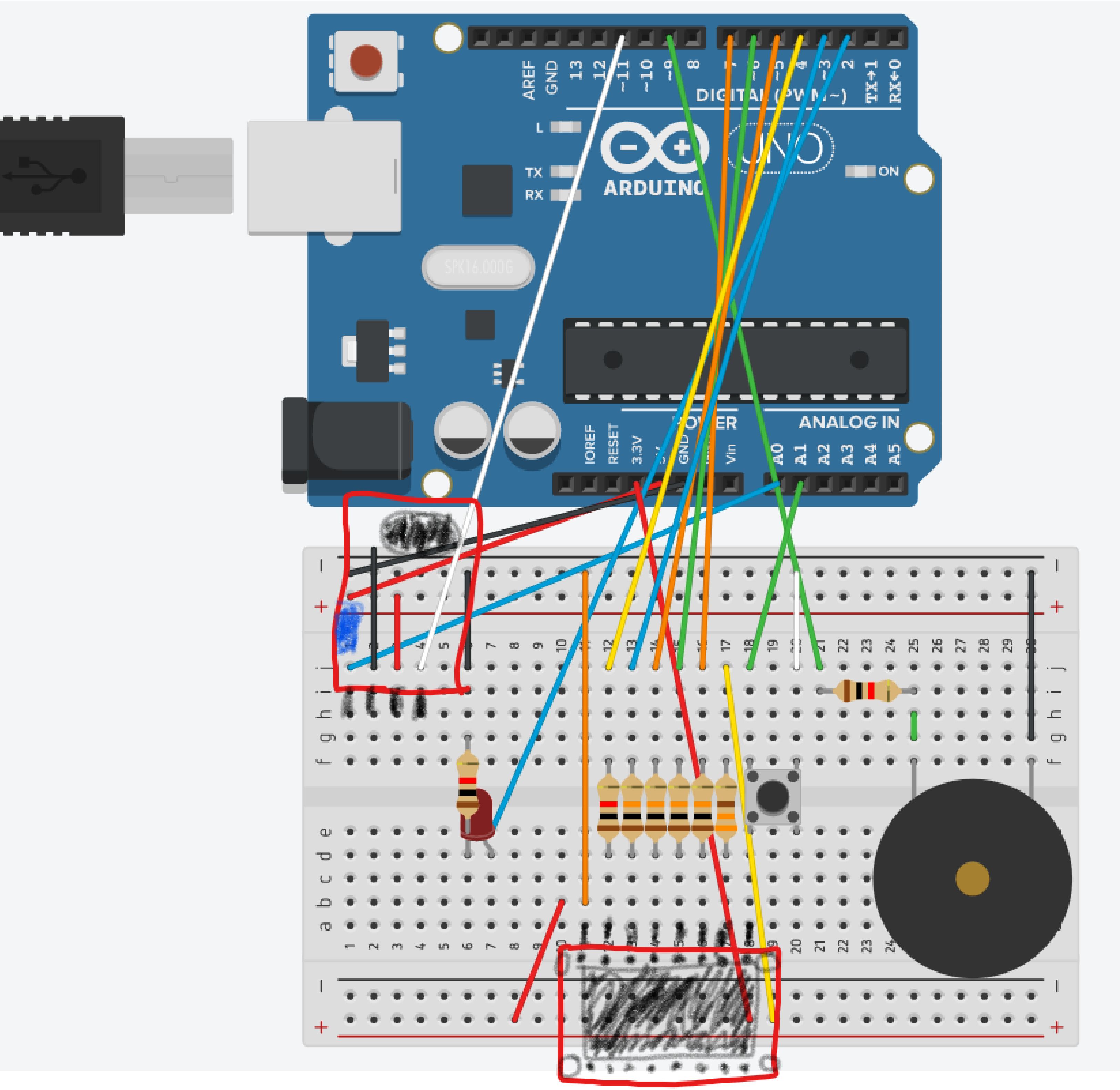
Hardware Components Used

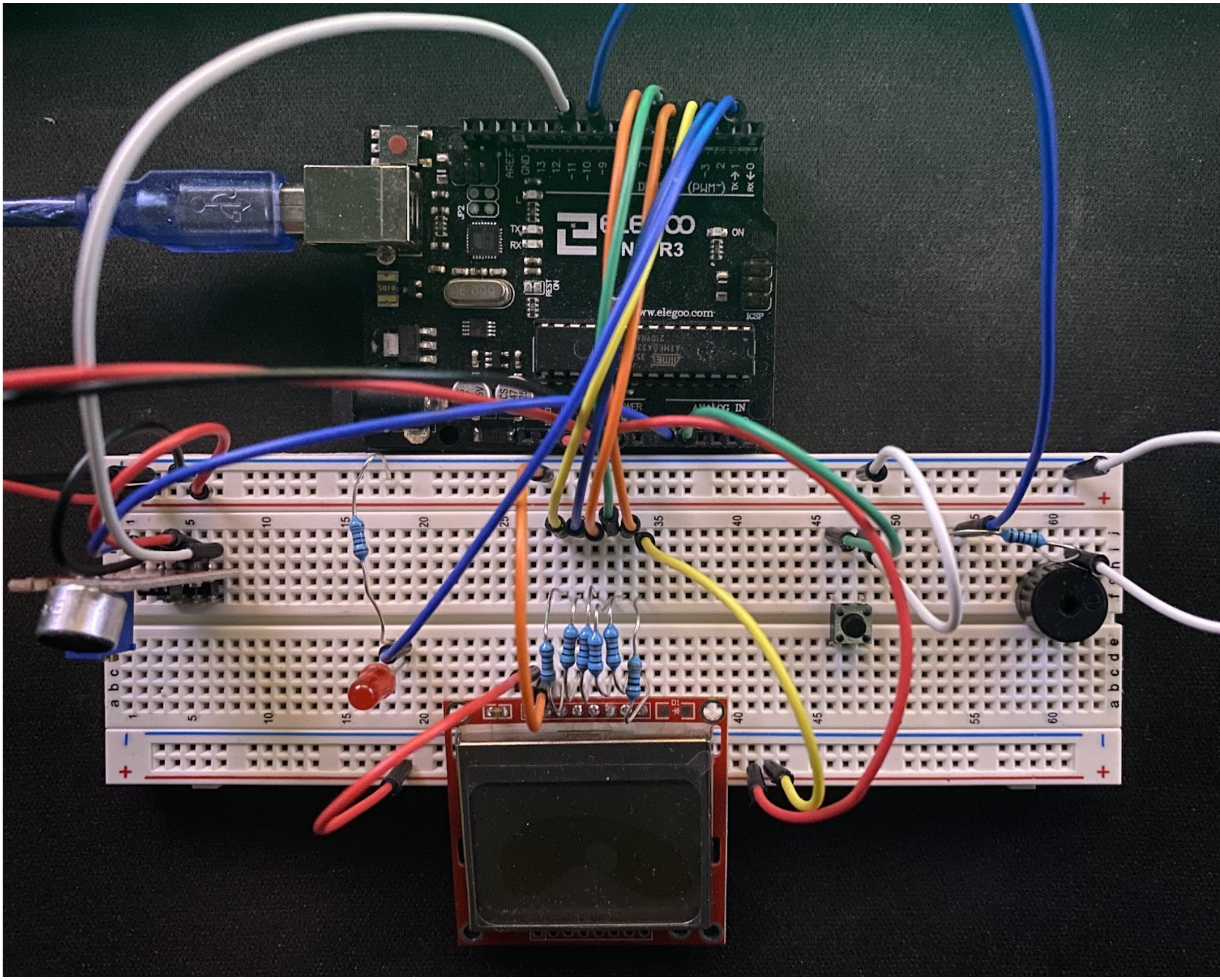
- Uxcell Microphone Sound Sensor
- Passive Buzzer
- Nokia 5110 LCD
- LED
- Resistors (of varying Ohm levels)
- Button
- Jumper Wires

Software Libraries Used

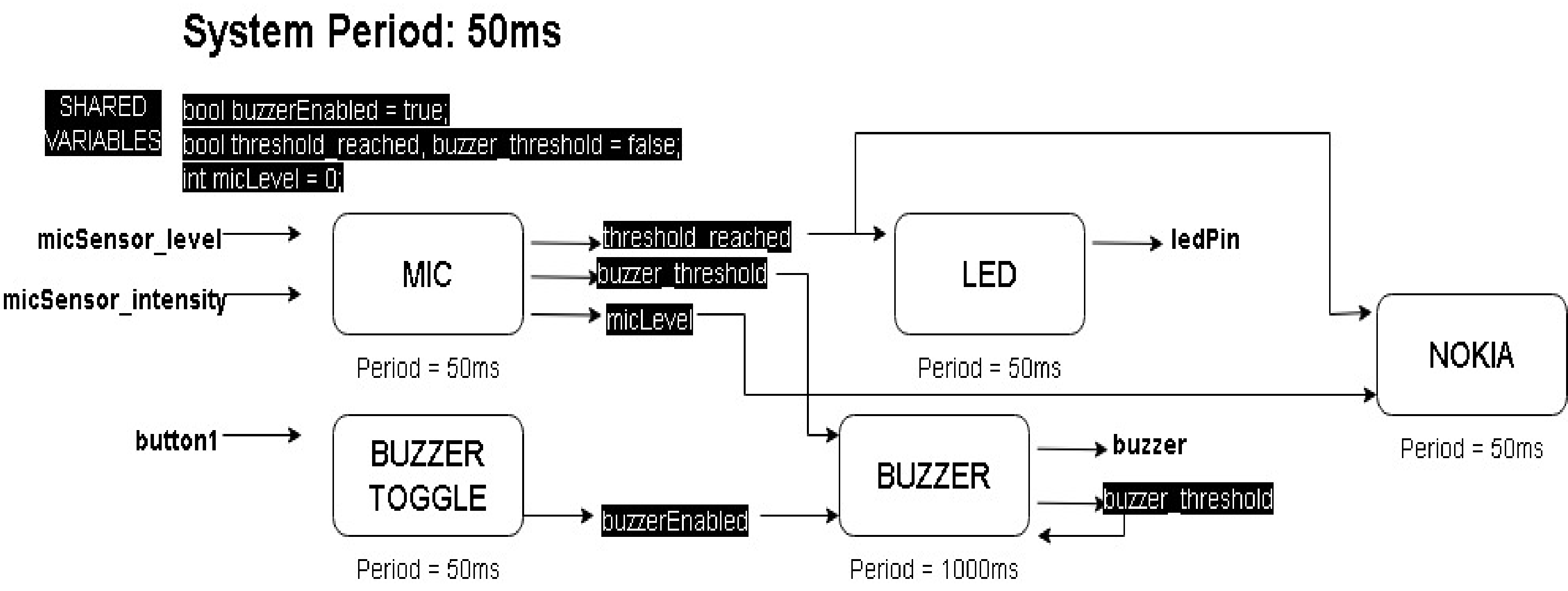
- Adafruit PCD8544 Library - For utilizing the Nokia 5110 LCD display and streamlining tasks by hiding many of the complexities under more simple commands.
    - Used library to initialize display, set contrast, clear buffer, and output corresponding text from microphone input level at various locations on the display.
- Adafruit GFX Library - Used in conjunction with PCD library to handle any displaying graphics.
- SPI Library - For communicating with SPI devices (needed as part of the Nokia 5110LCD implementation).

Wiring Diagram

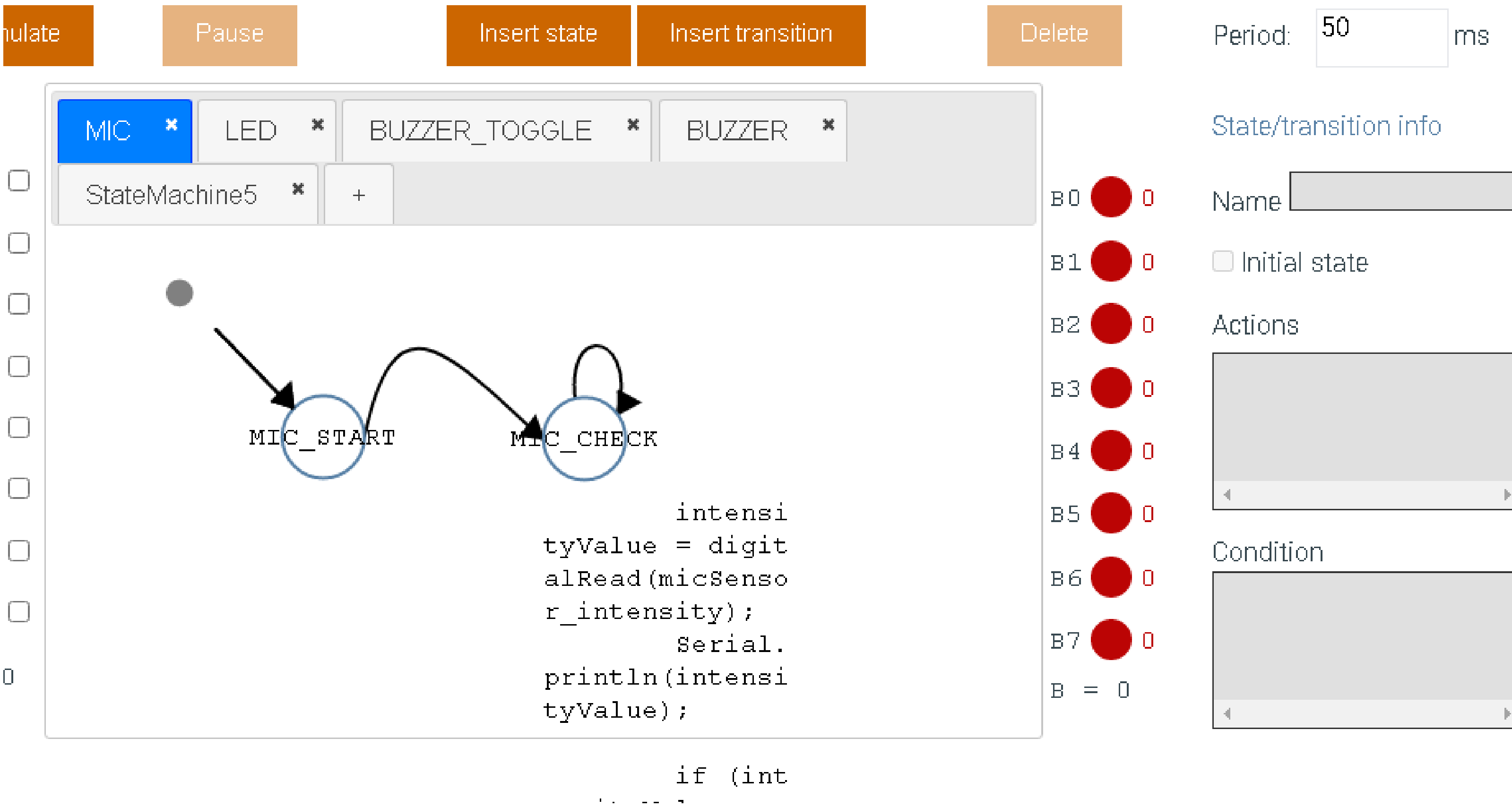




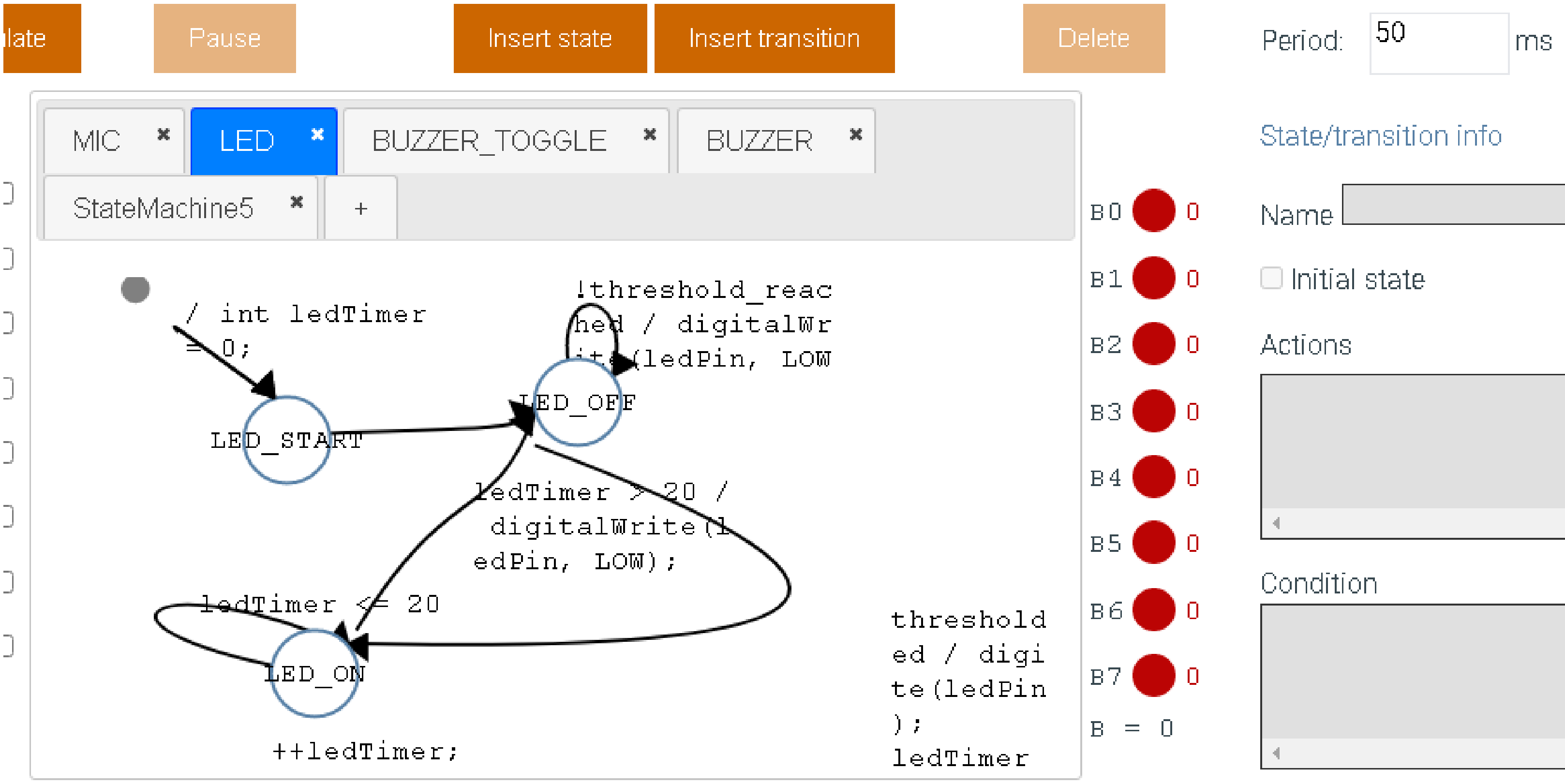
Task Diagram



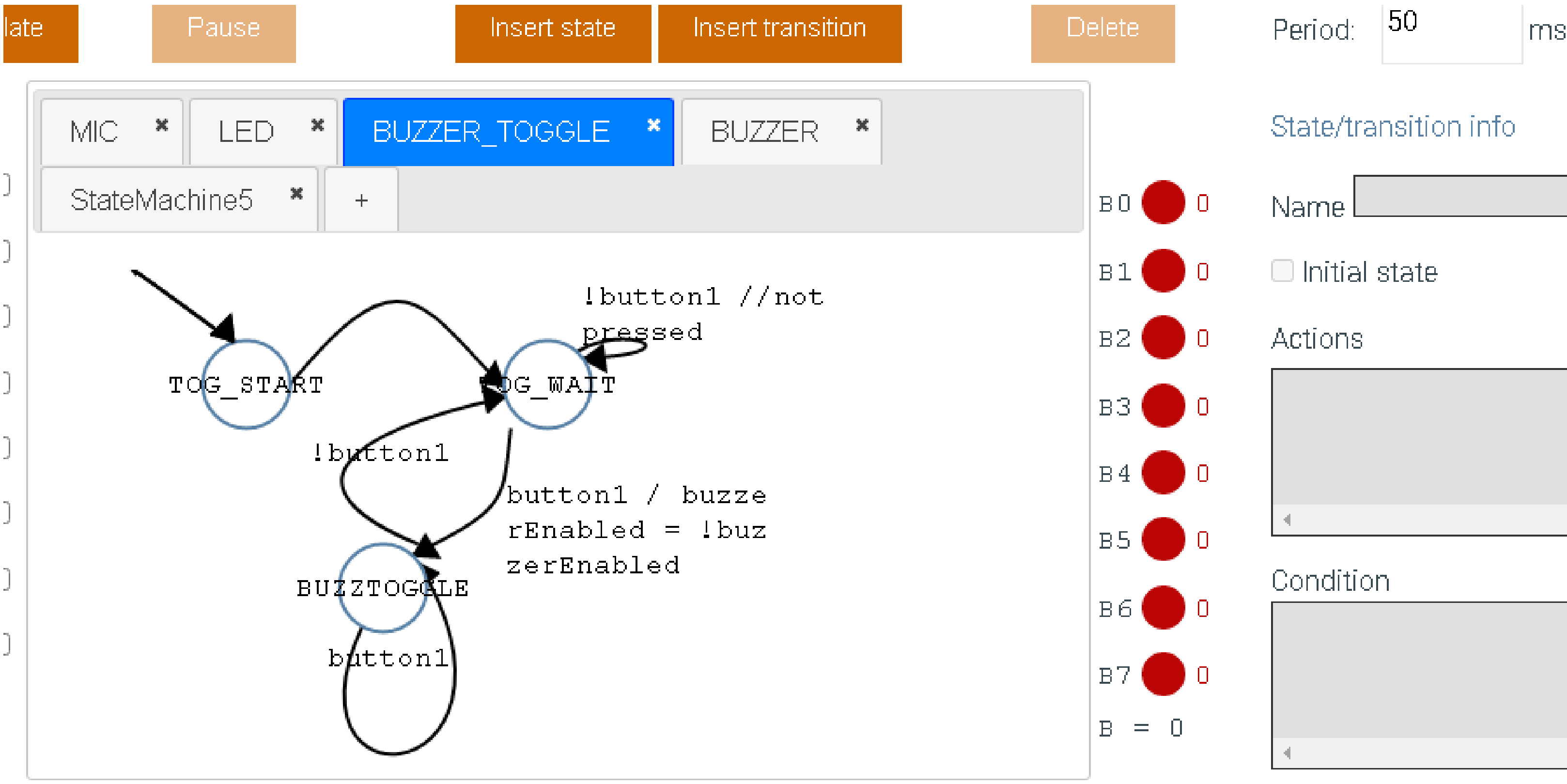
SynchSM Diagrams



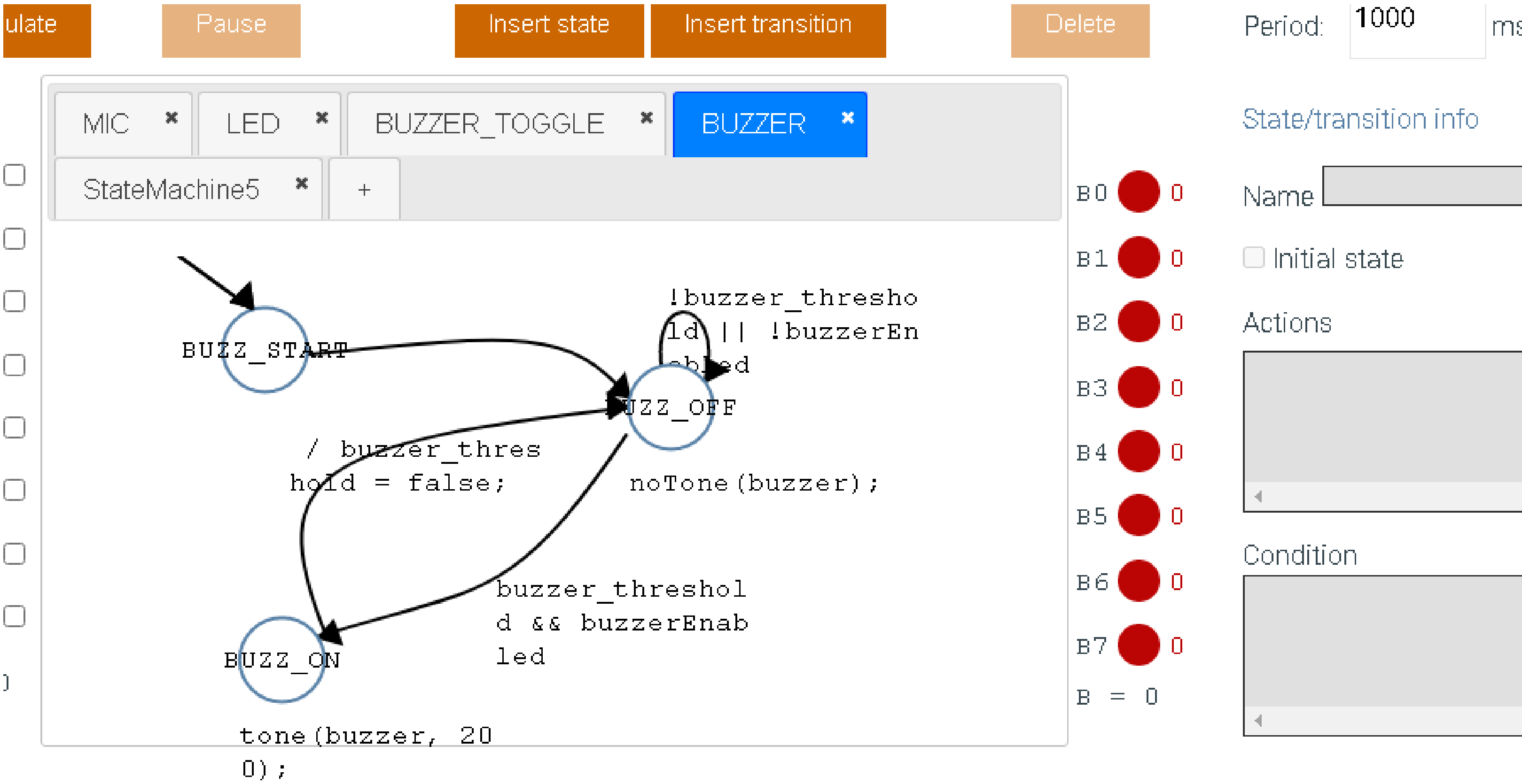
```
                    intensi
tyValue = digit
alRead(micSenso
r_intensity);
          Serial.
println(intensi
tyValue);

          if (int
```


late
Pause
Insert state
Insert transition
Delete
MIC ✖
LED ✖
BUZZER_TOGGLE ✖
BUZZER ✖
StateMachine5 ✖
+
/ int ledTimer
= 0;
LED_START
!threshold_reac
hed / digitalWr
ite(ledPin, LOW
LED_OFF
ledTimer > 20 /
digitalWrite(l
edPin, LOW);
ledTimer <= 20
LED_ON
++ledTimer;
threshold
ed / digi
te(ledPin
);
ledTimer
B0 ● 0
B1 ● 0
B2 ● 0
B3 ● 0
B4 ● 0
B5 ● 0
B6 ● 0
B7 ● 0
B = 0
State/transition info
Name
□ Initial state
Actions
Condition

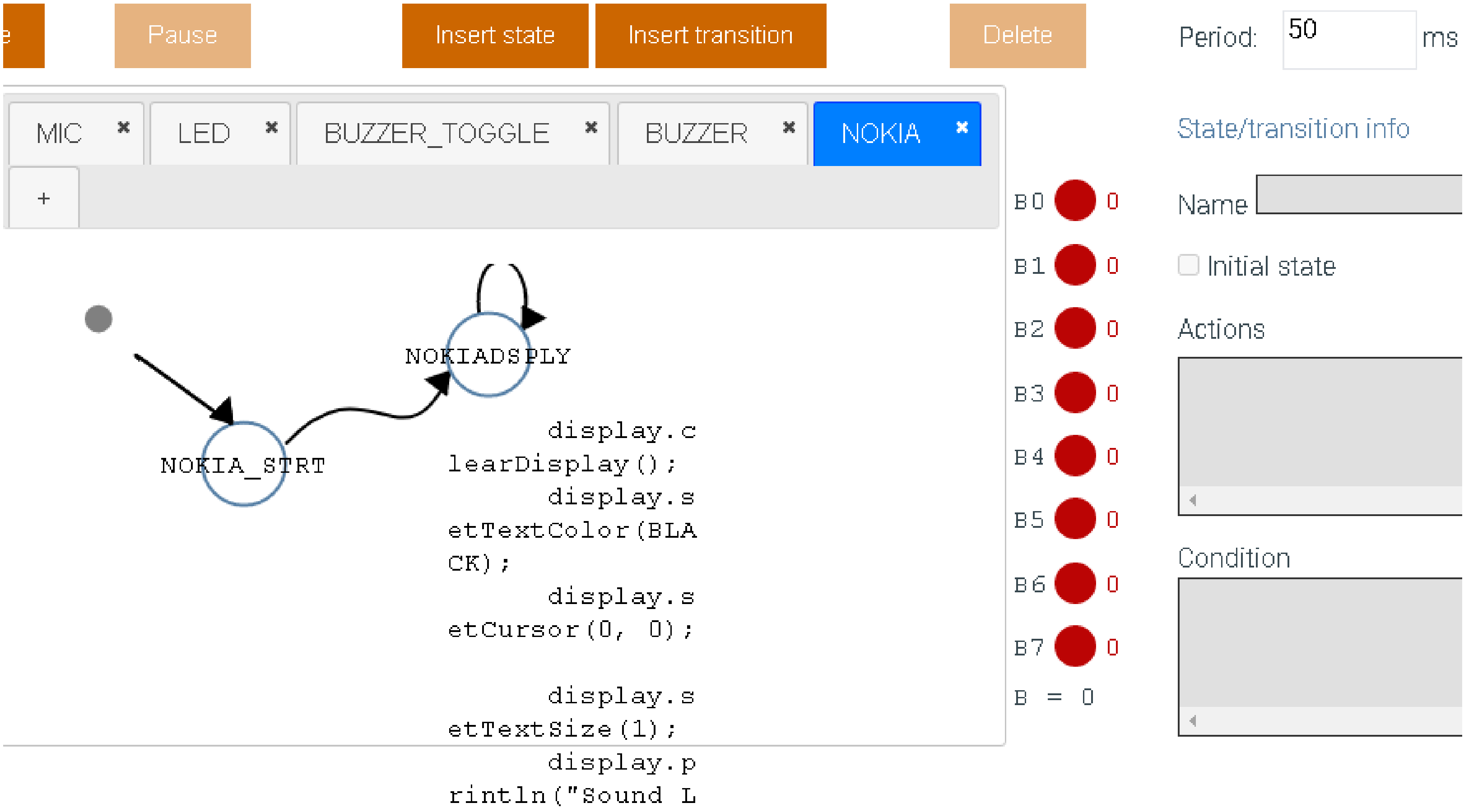
Period: 50 ms



Period: 50 ms



Period: 1000 ms



Period: 50 ms

```
        display.c
learDisplay();
        display.s
etTextColor(BLA
CK);
        display.s
etCursor(0, 0);

        display.s
etTextSize(1);
        display.p
rintln("Sound L
```