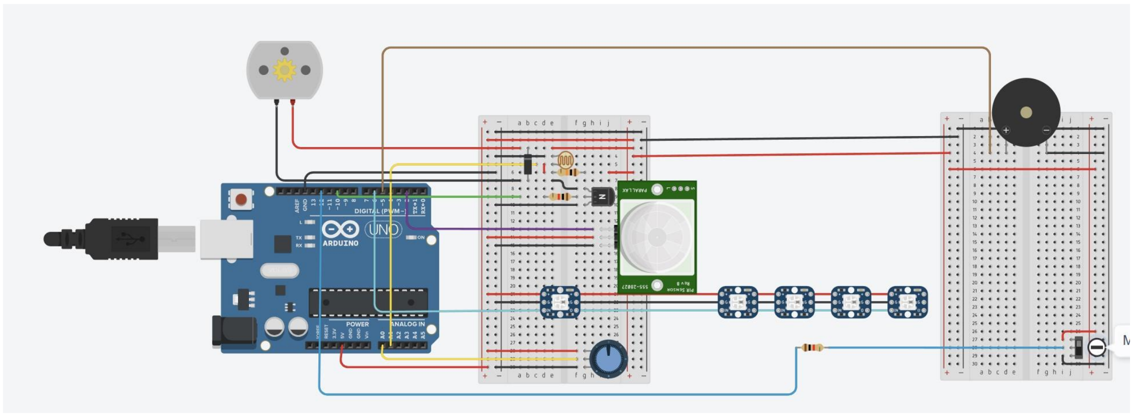
Department of Electronic Engineering
Royal Holloway, University of London
EE1000: Creative Engineering Team Project 1

Formal report
NAME: Aymen Mtibaa, year 1, 20/10/23

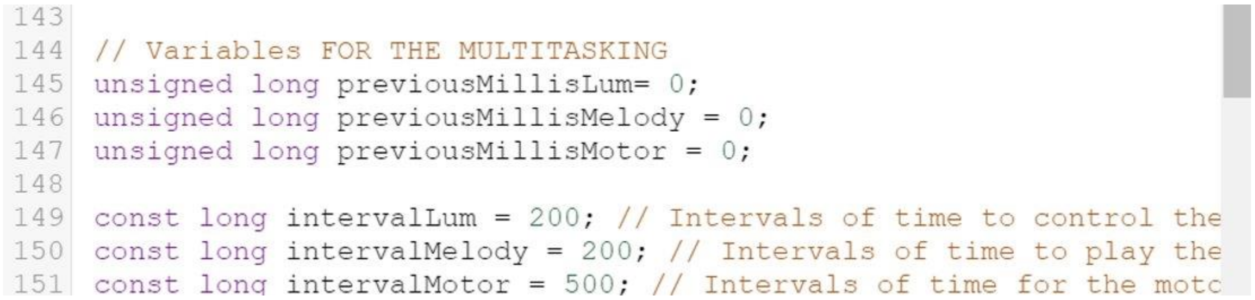
In this report, I will recount our recent group project, while talking about my role as a software engineer in creating an interactive, sensor-based device for babies within a four-week deadline.

As a group, we were tasked with developing a product that adhered to specific criteria yet allowed for creative freedom. The concept we agreed upon was inspired by an image, featuring stars that rotated with accompanying music – a soothing presence for a baby. My responsibility as the software engineer was pivotal, involving intricate coding and system integration.

The first week was dedicated to layout and design in Tinkercad. My role involved meticulously organizing the digital blueprint, ensuring clarity in pin assignments for seamless coding. The ambition behind my code was not just functional but also efficient; it was designed to activate only in the presence of a baby, detected via a PIR sensor, to conserve battery life. The device would also assess the room's brightness and the baby's state – whether they were awake and crying – to activate lights and music.



One of the most challenging aspects was coding the simultaneous operation of the DC motor, piezo, and neopixels. Achieving this required a clever programming strategy where each element was given 200 milliseconds of operation time within a continuous loop, creating the illusion of simultaneous activity.

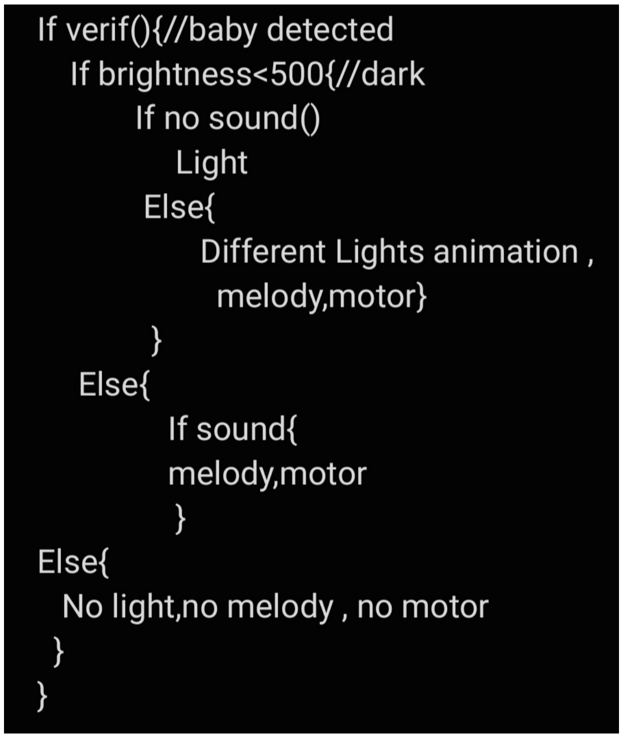


```

// Variables FOR THE MULTITASKING
unsigned long previousMillisLum= 0;
unsigned long previousMillisMelody = 0;
unsigned long previousMillisMotor = 0;

const long intervalLum = 200; // Intervals of time to control the
const long intervalMelody = 200; // Intervals of time to play the
const long intervalMotor = 500; // Intervals of time for the motc
```

As the project progressed, I focused on refining the code. The logic was straightforward yet effective: if a baby was detected, the system would check for darkness to activate lights, then assess if the baby was crying to play music and rotate the stars. We were mindful not to disturb a sleeping baby, ensuring the device's operation was sensitive to the baby's needs.



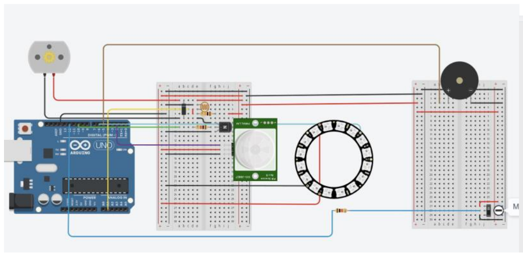
```
If verif(){//baby detected
   If brightness<500{//dark
         If no sound()
             Light
          Else{
               Different Lights animation ,
                 melody,motor}
           }
    Else{
            If sound{
            melody,motor
             }
Else{
  No light,no melody , no motor
 }
}
```

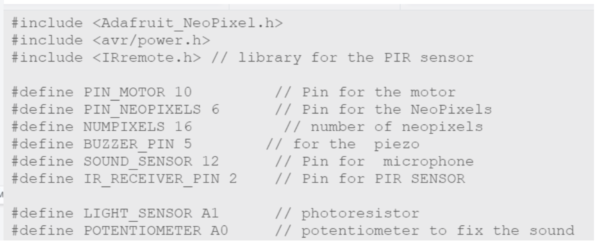
(This summary provides an overview of the code,

distilling its key functions and logic for a clearer understanding,

as including the entire 700-line code would be impractical)

In the final stages, we made two significant changes: removing the potentiometer, as the default sound level was deemed perfect, and upgrading to a 16-neopixel ring for enhanced lighting. These decisions, driven by budget considerations and aesthetic impact, required minimal code adjustments such as changing the number of Neopixels to 16 instead of 5 in the code.



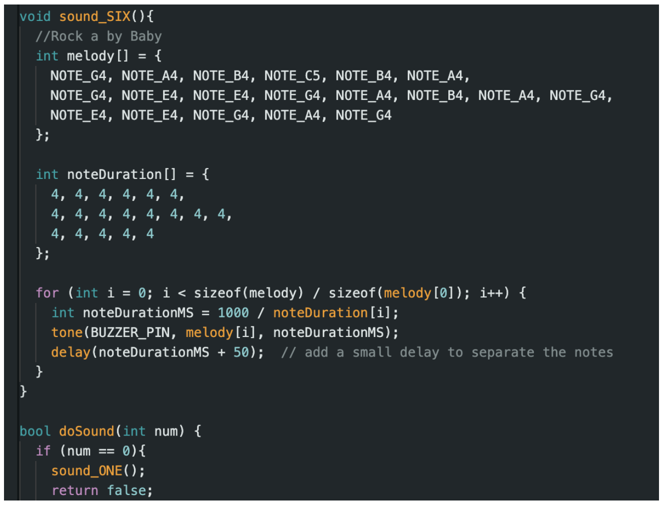


```
#include <Adafruit_NeoPixel.h>
#include <avr/power.h>
#include <IRremote.h> // library for the PIR sensor

#define PIN_MOTOR 10         // Pin for the motor
#define PIN_NEOPIXELS 6      // Pin for the NeoPixels
#define NUMPIXELS 16          // number of neopixels
#define BUZZER_PIN 5        // for the  piezo
#define SOUND_SENSOR 12      // Pin for  microphone
#define IR_RECEIVER_PIN 2    // Pin for PIR SENSOR

#define LIGHT_SENSOR A1      // photoresistor
#define POTENTIOMETER A0     // potentiometer to fix the sound
```

Choosing the final song for the device was deferred until the end, replacing the repetitive 'Happy Birthday' tune used during testing. This change, though small, was crucial in enhancing the user experience.



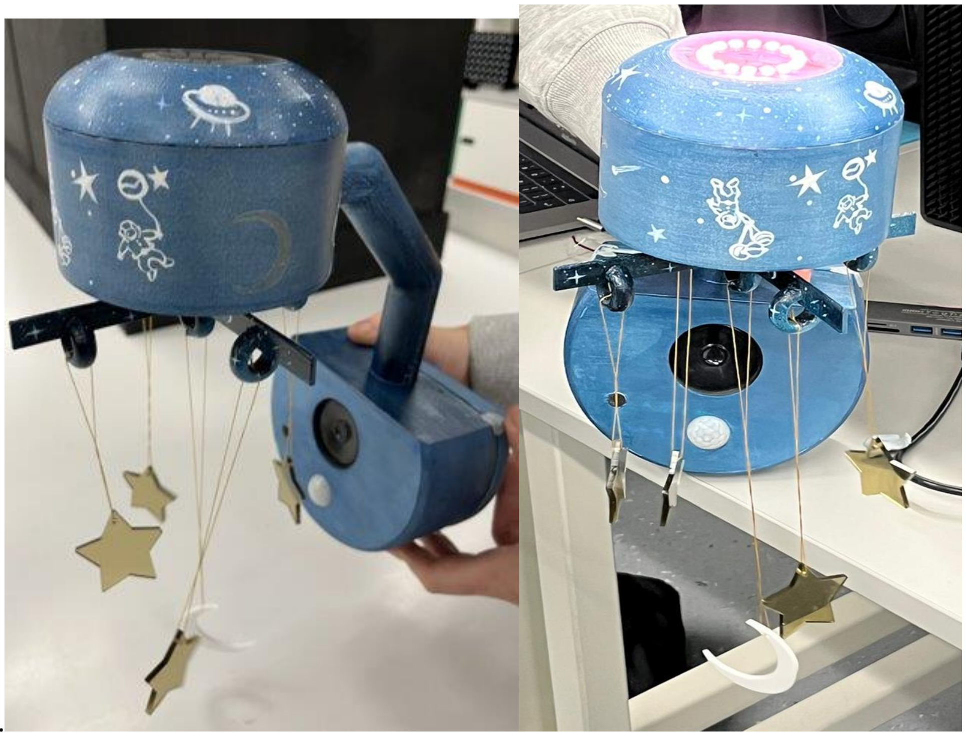
```
void sound_SIX(){
  //Rock a by Baby
  int melody[] = {
    NOTE_G4, NOTE_A4, NOTE_B4, NOTE_C5, NOTE_B4, NOTE_A4,
    NOTE_G4, NOTE_E4, NOTE_E4, NOTE_G4, NOTE_A4, NOTE_B4, NOTE_A4, NOTE_G4,
    NOTE_E4, NOTE_E4, NOTE_G4, NOTE_A4, NOTE_G4
  };

  int noteDuration[] = {
    4, 4, 4, 4, 4, 4,
    4, 4, 4, 4, 4, 4, 4, 4,
    4, 4, 4, 4, 4
  };

  for (int i = 0; i < sizeof(melody) / sizeof(melody[0]); i++) {
    int noteDurationMS = 1000 / noteDuration[i];
    tone(BUZZER_PIN, melody[i], noteDurationMS);
    delay(noteDurationMS + 50);  // add a small delay to separate the notes
  }
}

bool doSound(int num) {
  if (num == 0){
    sound_ONE();
    return false;
```

In conclusion, this project was not only a journey in software engineering but also a lesson in teamwork, time management, and the art of balancing functionality with creativity. The stress, while sometimes overwhelming, fueled our progress and innovation



.