****UNIVERSITY OF BOHOL

Tagbilaran City, Bohol, Philippines

COLLEGE OF ENGINEERING, TECHNOLOGIES,

ARCHITECTURE AND FINE ARTS

COMPUTER ENGINEERING

OBJECT ORIENTED PROGRAMING

CPEP121A

WEB SYSTEMS AND TECHNOLOGIES

CS123T

**SUMOBOT / HOCKEYBOT**

**Submitted by:**

ORIG, JEFF AXEL P.

**Introduction**

This project combines two robotics modes in a single system: Sumobot and Hockeybot. In Sumobot mode, the robot operates autonomously to detect and push opponents out of a ring using sensor-based logic. In Hockeybot mode, the robot can be remotely controlled via Bluetooth, allowing users to maneuver it like a player in a robotic hockey game. The dual-mode functionality showcases both autonomous behavior and real-time manual control, making it a versatile platform for robotics experimentation, competition, and fun.

**Scope and Limitation**

**Scope**

This project focuses on developing a dual-mode mobile robot that can function both autonomously and via Bluetooth control. In Sumobot mode, the robot uses sensors to detect edges and opponents, enabling it to compete in sumo-style matches without human intervention. In Hockeybot mode, the robot responds to Bluetooth commands from a custom mobile application, allowing users to control its movements for interactive play. The system demonstrates basic robotics concepts, sensor integration, motor control, and wireless communication.

**Limitations**

* The autonomous behavior in Sumobot mode is limited to simple push-and-avoid logic and may not adapt to highly complex or dynamic opponents.
* Bluetooth range restricts the effective control distance in Hockeybot mode.
* The system does not support simultaneous dual-mode operation; it must be toggled manually.
* Performance may vary based on surface conditions, lighting (for IR sensors), and battery level.

**PROBLEM REQUIREMENTS**

**Purpose**

The purpose of this project is to design and develop a multifunctional robot that can operate in two distinct modes—**Sumobot** and **Hockeybot**—to demonstrate both autonomous decision-making and remote-controlled navigation. It aims to enhance understanding of robotics concepts such as sensor-based automation, motor control, wireless communication, and user interface integration. This project also serves as a platform for learning, competition, and entertainment by allowing users to explore real-world applications of embedded systems and robotics engineering.

**Overall Description**

The Sumobot / Hockeybot is a dual-function mobile robot designed for both autonomous and manual operation. In Sumobot mode, the robot uses infrared sensors and strategic programming to detect opponents and ring boundaries, enabling it to engage in robotic sumo matches by pushing opponents out of the ring while avoiding self-elimination. In Hockeybot mode, it becomes a Bluetooth-controlled vehicle, allowing real-time user interaction through a mobile application to simulate a robotic hockey game.

The system is built on a microcontroller platform (such as Arduino), equipped with sensors, DC motors, and a Bluetooth module. A toggle mechanism (such as a physical switch) enables users to change between modes. The project emphasizes modularity, versatility, and educational value, providing hands-on experience in both autonomous robotics and wireless control systems.

**Analysis**

**Input requirements**

* Mode Selector Switch: A physical switch (e.g., connected to a digital pin) to toggle between Sumobot and Hockeybot modes.
* IR Sensors: Used in Sumobot mode to detect line boundaries and opponent presence.
* Bluetooth Module (e.g., HC-05): Receives directional commands from the mobile app in Hockeybot mode.
* Power Supply: Batteries or USB to provide power to the microcontroller and motors.
* Motor Driver Input: Signals from the microcontroller to control motor speed and direction.

**Output requirements**

* Motor Movement: The robot should move forward, backward, turn, or stop based on sensor readings (Sumobot) or Bluetooth commands (Hockeybot).
* Bluetooth Response (optional): Acknowledgement messages sent back to the app (if implemented).
* Autonomous Behavior: In Sumobot mode, the robot should make real-time decisions to stay in the ring and push opponents.

**Necessary Formula and their Description**

While the project is primarily hardware- and logic-driven, a few basic formulas are used in interpreting sensor data and controlling motor speed and direction:

1. **Ultrasonic Distance Calculation**

return duration \* 0.034 / 2;

*Description:*  
 This formula calculates the distance of an object using the time it takes for an ultrasonic pulse to travel to an object and back.

* duration: time in microseconds (µs) from pulse to echo.
* 0.034 cm/µs: approximate speed of sound.
* Dividing by 2 accounts for the round-trip (to object and back).

1. **PWM Motor Speed Control**

*Description:*  
 The motors are controlled using PWM values (0–255) via analogWrite(). A higher value means faster speed. For example:

* analogWrite(pin, 255) → 100% speed
* analogWrite(pin, 128) → ~50% speed

1. **IR Sensor Line Detection Logic**

if (digitalRead(IR\_SENSOR) == LOW)

*Description:*

The IR sensor returns LOW when it detects the edge (e.g., white surface of the ring), triggering avoidance (reverse). This isn't a numerical formula, but it is essential for defining robot behavior logic in autonomous mode.

1. **Delay-Based Motion Timing**

delay(ms);

*Description:*  
 Used for timing actions like reversing or attacking. For example, delay(400) means the robot will maintain its current action (e.g., reversing) for 400 milliseconds before reevaluating behavior.

1. **Time-Based Conditional Monitoring**

while (millis() - startTime < duration)

*Description:*

This checks elapsed time in milliseconds to control how long the robot continues a specific action (like attack) and keeps monitoring for obstacles using the IR sensor.

**Design**

**Files and Their Descriptions (Embedded System Perspective)**

The Sumobot / Hockeybot program is implemented using the Arduino IDE, structured in a single .ino sketch file. This file contains all functions related to both Autonomous (Sumo) and Bluetooth-Controlled (Hockey) modes:

* Setup and Loop Functions: Initialize pins and determine current mode.
* Autonomous Mode Functions: Include autonomousMode(), attack(), reverse(), and search() to handle logic for sumo battles using sensors.
* Controlled Mode Functions: Include controlledMode() and directional movement functions (moveForward(), spinLeft(), etc.) triggered by Bluetooth commands.
* Shared Motor Control: Modular functions control both wheels using analogWrite() and digitalWrite().

**User Interface Design (MIT App Inventor - Hockeybot Controller)**

The mobile app was developed using MIT App Inventor to control the robot via Bluetooth. The UI design includes:

* BluetoothClient Component: Enables pairing and communication with the robot’s HC-05 module.
* Buttons:
  + Forward (W)
  + Backward (S)
  + Turn Left Gradually (A)
  + Turn Right Gradually (D)
  + Spin Left (L)
  + Spin Right (R)
  + Gas/Boost (G)

Touch-Up Handling (App Logic)

The app sends movement commands on button press and a stop command on release, enabling smooth, joystick-like control via Bluetooth.

**Features of the Project**

* Dual-Mode Functionality: Switch between autonomous Sumobot mode and manual Hockeybot mode using a hardware switch.
* Sensor Integration: Uses an ultrasonic sensor to detect opponents and an IR sensor to avoid ring edges.
* Bluetooth Control: A custom MIT App Inventor app allows directional control with responsive touch-based commands.
* Boost Function: Increases motor speed temporarily for faster movement.
* Modular Code Structure: Organized motor, sensor, and behavior functions for easier maintenance.

**Implementation**

**URL of the Saved Source Code**

<https://github.com/046289/Sumobot-Hockeybot>

**Software Package Files and their Descriptive Information**

* MIT App Inventor Project (.aia / .apk):

Used to create the mobile Bluetooth controller app. It includes buttons for directional commands (forward, backward, left, right, spins, and gas), and uses the BluetoothClient component to send commands to the robot. Touch-up events are implemented to send a stop signal when buttons are released, ensuring responsive, joystick-like control.

* Arduino Sketch (.ino):

Contains the main logic for the robot, including motor control, Bluetooth communication, and mode switching. It listens for incoming Bluetooth commands and executes corresponding movements, or runs autonomous sumo logic when in that mode.

**Function Declarations and their Descriptive Purposes**

1. **Switch Modes:**

Purpose: Toggles between autonomous mode (Sumobot) and Bluetooth-controlled mode (Hockeybot).

1. **Motor Control Functions:**

Purpose: Functions like moveForward(), turnLeft(), stopMotors() allow the robot to perform various movements either autonomously or based on Bluetooth commands.

1. **Sensor Handling:**

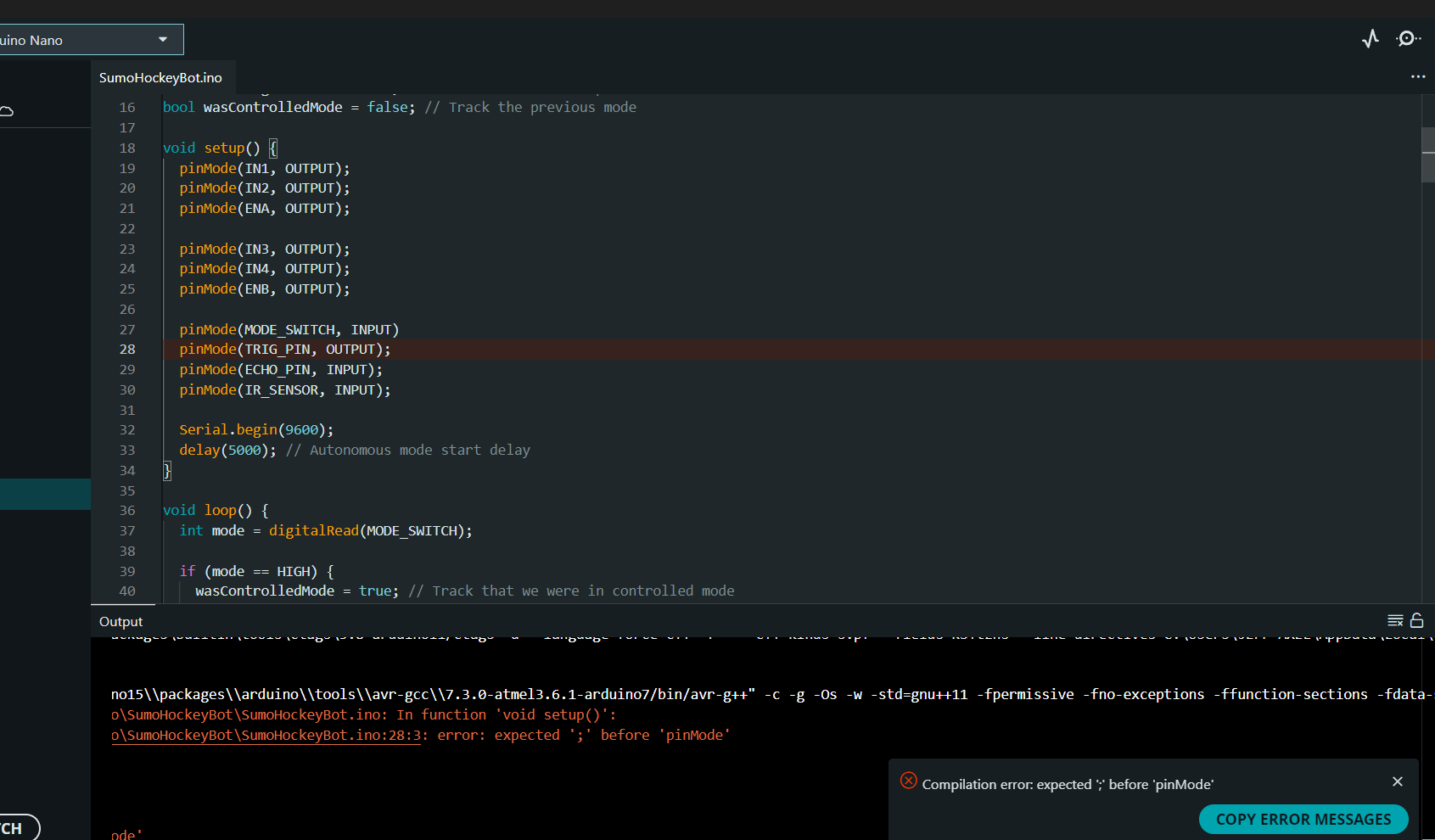
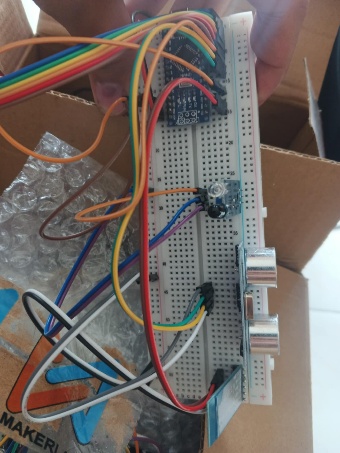
Purpose: Functions like readUltrasonic() and detectIR() continuously monitor the environment, enabling the robot to navigate and perform actions like attacking or avoiding obstacles.

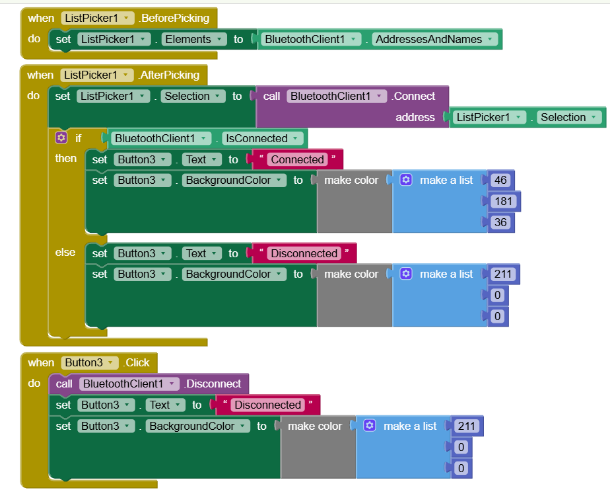
1. **Bluetooth Communication Functions:**

Purpose: Handles incoming Bluetooth commands, ensuring responsive control via an app (MIT App Inventor).

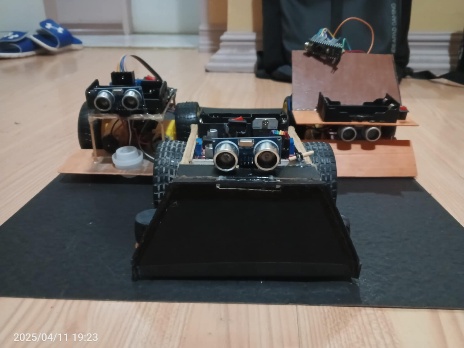
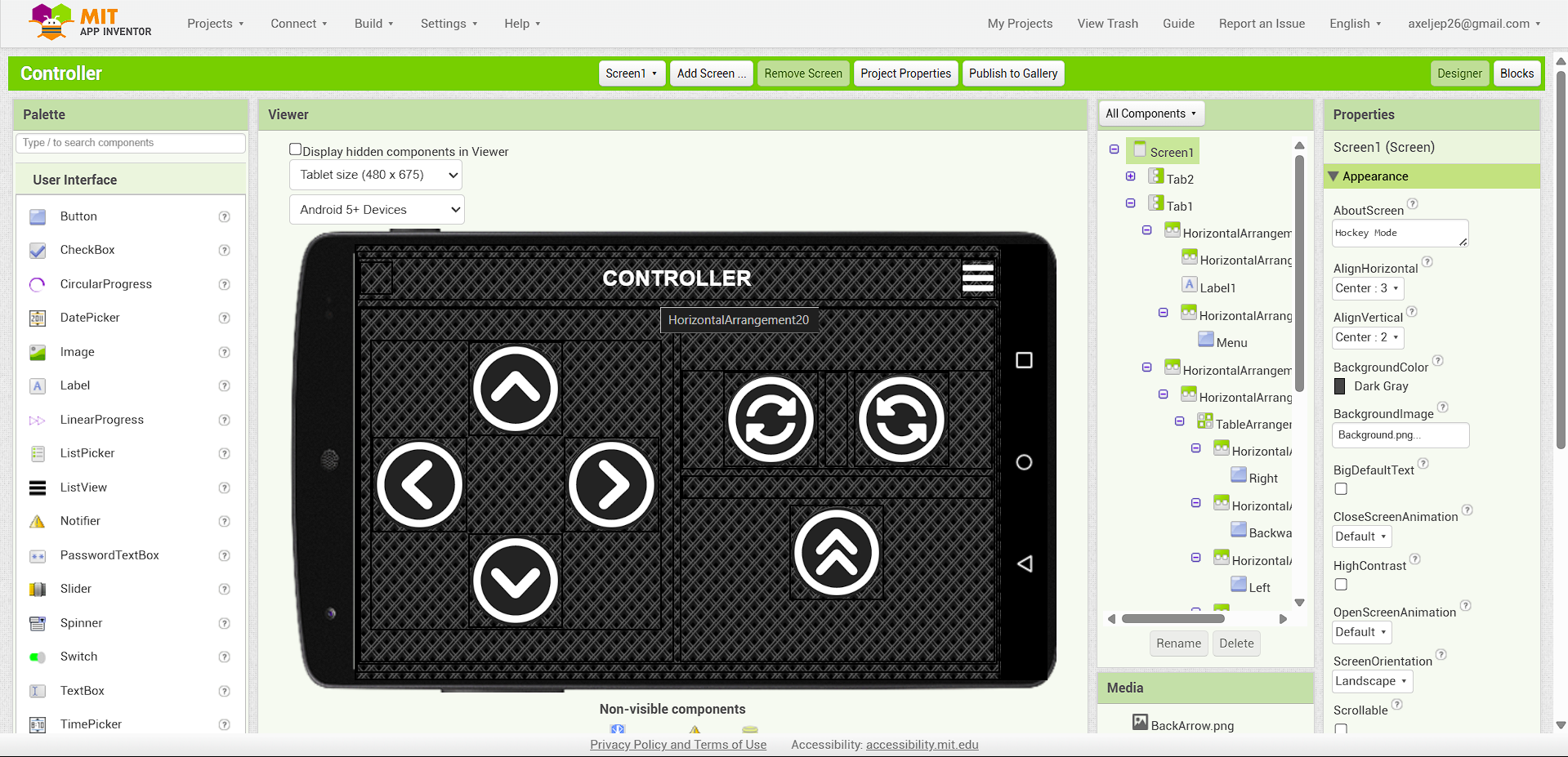
**Testing and debugging**

Initial output.





Final output.

**** ****

**The Creator**

Jeff Axel P. Orig - Handled all hardware and software tasks—from wiring and configuring sensors to developing the Arduino code and Bluetooth controller using MIT App Inventor.

**UNIVERSITY OF BOHOL**

CITY OF TAGBILARAN

**CURRICULUM VITAE**

**I. PERSONAL INFORMATIONS**

**Name :** Jeff Axel P. Orig

**Address :** Pamaong Stanai Rd.,

brg. Cogon, Tagbilaran City

**Birthdate :** June 26, 2005

**Nationality :** Filipino

**Religion :** Roman Catholic

**Civil Status :** Single

**Parents :** Eleazar Wilson Orig

Juliet Orig

**II. EDUCATIONAL BACKGROUND**

**Tertiary :** University of Bohol, College of Engineering,

Technology, Architecture and Fine Arts

Dr. Cecilio Putong Street, Cogon Tagbilaran City, Bohol

2023-Present

**Secondary :** Cor Jesu Institute of Mabini Inc.

Mabini, Davao de Oro

S.Y. 2017-2023

**Elementary :** Maabini Central Elementary School

Mabini, Davao de Oro

S.Y. 2011-2017

**Photos**

**Future Development**

**Recommendations**

To improve the functionality and user experience of the Sumo/Hockey Bot, it is recommended to integrate additional sensors such as infrared or ultrasonic for edge and obstacle detection in Sumo mode. For Hockey mode, using a physical joystick can make the operation more intuitive. Adding Bluetooth feedback or status indicators (e.g., battery level, connection status) could also enhance usability. These upgrades aim to improve responsiveness, safety, and interactivity.

**Project Cost**

|  |  |
| --- | --- |
| Expenses | Costs |
| L298N Motor Driver | 75 |
| DC Geared Motor 2x | 130 |
| Ultrasonic Sensor | 49 |
| Ir Sensor 2x | 50 |
| DC-DC Booster 2x | 110 |
| Battery 2x | 370 |
| Battery Holder | 49 |
| Bluetooth Module | 199 |
| Arduino Nano | 187 |
| Shipping fee | 280 |
| Printing | 110 |
| Bookbinding | 150 |
| Total | 1,759 |

**Glossary**

**Arduino:**

An open-source electronics platform based on easy-to-use hardware and software. It is used to control motors and sensors in the robot.

**MIT App Inventor:**

A visual programming environment used to create Android apps, such as the Bluetooth controller for the bot.

**Bluetooth Module (e.g., HC-05/HC-06):**

A wireless communication module that allows the Arduino to receive commands from the mobile app.

**Hockey Mode:**

A manual control mode where the user directs the robot via Bluetooth commands from the mobile app.

**TouchDown/TouchUp Events:**

Actions in the app where a command is sent when a button is pressed (TouchDown) and a stop signal when released (TouchUp), mimicking joystick behavior.

**DC Motor Driver (L298N):**

A module used to control the speed and direction of DC motors in the robot.

**Sumo Mode:**

An autonomous mode where the robot uses sensors to detect opponents and stay within the ring.

**Sketch (.ino file):**

The Arduino program file that contains the code for controlling the robot's behavior.

**Bibliography**

Mischianti, L. (2023, January 3). *Arduino Nano: High resolution pinout, datasheet, and specs*. Mischianti Projects. <https://mischianti.org/arduino-nano-high-resolution-pinout-datasheet-and-specs/>

MIT App Inventor. (n.d.). *Lesson 2.2-2: User-Interaction*. MIT App Inventor. <https://appinventor.mit.edu/explore/teach/module-2-2/user-interaction.html>