**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG – HCM**

A picture containing logo

Description automatically generated**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**Môn: Nhập môn lập trình điều khiển thiết bị thông minh**

**Lớp: 19\_23**

Họ tên sinh viên: Lê Mai Nguyên Thảo

MSSV: 19120661

TP. Hồ Chí Minh, ngày 15 tháng 1 năm 2023

**MỤC LỤC**

**LỜI MỞ ĐẦU 3**

**GIỚI THIỆU DỰ ÁN 4**

**CHUẨN BỊ 5**

**SCHEMATICS 8**

**TRIỂN KHAI DỰ ÁN 9**

1. **VỀ EDGE IMPULSE 9**
2. **THU THẬP DỮ LIỆU BẰNG EDGE IMPULSE 10**
3. **THIẾT KẾ IMPULSE 12**
4. **TRAINING MODEL 16**
5. **TEST MODEL 18**
6. **TRIỂN KHAI MÔ HÌNH LÊN THIẾT BỊ NHÚNG 19**
7. **FILE PROJECT ARDUINO 20**
8. **KẾT NỐI ARDUINO VỚI UNITY 23**

**ĐÁNH GIÁ TỔNG QUÁT 25**

**DEMO 26**

**LỜI MỞ ĐẦU**

Thời đại công nghệ ngày càng phát triển, theo đó, trí tuệ nhân tạo ngày càng được áp dụng vào đời sống. Từ thiết bị giải trí cho đến những vật dụng hàng ngày, đâu đâu cũng có thể đưa trí tuệ nhân tạo vào. Đó chính là AIoT – trí tuệ nhân tạo vạn vật. AIoT là sự kết hợp của các công nghệ trí tuệ nhân tạo (AI) với cơ sở hạ tầng Internet of Things (IoT) để đạt được các hoạt động IoT hiệu quả hơn, cải thiện tương tác giữa người và máy cũng như tăng cường quản lý và phân tích dữ liệu.

AIoT được dự đoán sẽ định nghĩa lại tương lai của tự động hóa công nghiệp và được cho là dẫn đầu cuộc cách mạng công nghiệp 4.0. Vì lẽ đó, môn học AIoT là một môn học rất bổ ích ở trường. Qua đó, sinh viên được tiếp cận với cách đưa AI vào đời sống thông qua các thiết bị điều khiển và cách nó vận hành trong từng dự án, trường hợp, lĩnh vực cụ thể.

Để hoàn thành được đồ án này, em xin chân thành cảm ơn các Thầy Cô đã hướng dẫn tận tình và có những bài học rất bổ ích. Trong quá trình thực hiện đồ án không thể tránh khỏi thiếu sót, em mong các Thầy Cô góp ý để em có thể hoàn thiện kỹ năng của mình hơn.

**GIỚI THIỆU DỰ ÁN**

Dự án em trình bày đến Thầy Cô hôm nay là dự án xây dựng thiết bị điều khiển game 3D bằng thiết bị MPU6050 và TinyML: 3D Unity game using MPU6050 & TinyML. Dự án dựa vào mô hình học máy nhận diện các cử chỉ liên tục: up, down, left, right để điều khiển xe tăng di chuyển tiến, lùi, quay trái, quay phải.

Để thực hiện được dự án này, em đã tham khảo từ rất nhiều nguồn như sau:

1. **Ý tưởng thực hiện**

Ý tưởng thực hiện dự án này em tham khảo hoàn toàn từ dự án 3D Unity game using MPU6050 & TinyML Tutorial của tác giả Vishwa. Tuy nhiên, trong quá trình thực hiện, em nhận thấy các bước cũng như hướng dẫn trong dự án này có quá sức với em nên em đã thực hiện dự án với những cách đơn giản hơn, tiện hơn.

Link dự án mẫu: [3D Unity game using MPU6050 and TinyML Tutorial - Hackster.io](https://www.hackster.io/theevildoof/3d-unity-game-using-mpu6050-and-tinyml-tutorial-8e3076)

1. **Xây dựng game 3D**

Để xây dựng được một game 3D cơ bản, em đã sử dụng phần mềm Unity. Vì là game cơ bản nên sẽ không tránh khỏi sai sót những lỗi logic. Tuy nhiên, trong phạm vi đồ án này, em sẽ không trình bày về cách xây dựng game 3D trên Unity mà sẽ tập trung vào cách xây dựng mô hình và triển khai vào thiết bị nhúng.

Link tham khảo xây dựng game 3d trên Unity: [Tanks - Unity Learn](https://learn.unity.com/project/tanks-tutorial)

1. **Thu thập data và train model**

Để thực hiện được đồ án này thì điều tất yếu phải làm đó là thu thập dữ liệu để xây dựng mô hình học máy. Toàn bộ quá trình từ xây dựng data cho đến train model và triển khai về thiết bị nhúng, em đều tham khảo ở link sau: [Gesture Classification with Esp32 and TinyML - Hackster.io](https://www.hackster.io/Yukio/gesture-classification-with-esp32-and-tinyml-dab252). Trong link này có hướng dẫn rất chi tiết các dòng lệnh để triển khai vào thiết bị nhúng.

1. **Cài đặt, kết nối ESP32 – CAM với Arduino IDE**

Để triển khai dự án vào thiết bị nhúng, em sử dụng ESP32 – CAM lập trình trên Arduino IDE. Và để có thể nhận thiết bị và nạp code, em đã kết nối ESP32 – CAM với phần mềm Arduino IDE. Quá trình này em tham khảo thông qua link: [(90) Cài Đặt Package ESP32 Trên Arduino IDE || How To Install ESP32 Package in Arduino IDE - YouTube](https://www.youtube.com/watch?v=7szltOyHOr0&list=PLYxM7t7aYNau4iwp1gJlBSFFwzBC9X_Wi&index=3).

1. **Kết nối Unity với Arduino**

Và dĩ nhiên, để có thể điều khiển được game Unity bằng thiết bị thì phải có sự kết nối giữa chúng. Để tạo được sự kết nối này, em tham khảo ở những link sau:

[(90) How to Connect Arduino to Unity? #1 Serial Communication between Arduino and Unity. - YouTube](https://www.youtube.com/watch?v=5ElKFY3N1zs&t=236s)

[(90) Ardity - free plugin for connecting Arduino & Unity - YouTube](https://www.youtube.com/watch?v=QeakrwahFBA&t=350s)

**CHUẨN BỊ**

Để thực hiện đồ án này, em sử dụng các thiết bị cũng như phần mềm / web online như sau:

1. **Thiết bị phần cứng**

**\*\* ESP32 – CAM AI – Thinker:**

ESP32 – CAM là một board mạch phát triển với chip điều khiển ESP32 – S, camera OV2640, khe cắm thẻ nhớ microSD và các GPIO để kết nối với thiết bị ngoại vi.

Đây cũng là board mạch để nạp code vào và thực thi.

**\*\* Cảm biến GY – 521 6DOF IMU MPU6050:**

MPU6050 là một trong những cảm biến chuyển động đầu tiên trên thế giới có tới 6 trục (mở rộng tới 9 trục) cảm biến tích hợp trong 1 chip duy nhất. MPU6050 tích hợp 6 trục cảm biến:

* Con quay hồi chuyển 3 trục (3 – axis MEMS gyroscope).
* Cảm biến gia tốc 3 chiều (3 – axis MEMS accelerometer).

**\*\* Điện trở 1/4W 1% 200R:**

Các thông số của điện trở:

* Giá trị điện trở 200R.
* Công suất 1/4W.
* Sai số 1%.
* Nhiệt độ hoạt động: -55 -> 150C.
* Cấu tạo màng kim loại.

Điện trở 200R 1/4W 1% phù hợp với các mạch điện tử cần độ chính xác, hoạt động ổn định, bền với thời gian. Điện trở cũng góp phần khiến mạch sẽ tự ngắt nếu dòng điện quá tải.

**\*\* Solderless Breadboard:**

Solderless Breadboard được dùng để tạo mẫu (cắm dây) các mạch điện tử không sử dụng mỏ hàn.

**\*\* Jumper Wires:**

Để các linh kiện, thiết bị kết nối được với nhau thì không thể thiếu Jumper Wires. Có 3 loại Jumper Wires: đực – cái, cái – cái, đực – đực.

**\*\* USB UART CH340G:**

Mạch chuyển USB UART CH340G là mạch chuyển tín hiệu cổng USB từ máy tính sang dạng nối tiếp dùng để nạp cho các mạch arduino hoặc dùng để giao tiếp giữa các board vi điều khiển với máy tính.

1. **Phần mềm / web online**

**\*\* Edge Impulse:**

Edge Impulse là web miễn phí cho phép xây dựng mô hình học máy một cách nhanh chóng. Edge Impulse có bộ Document cũng như hướng dẫn khá đầy đủ nên rất dễ sử dụng. Ngoài ra, Edge Impulse còn hỗ trợ triển khai mô hình học máy về thiết bị nhúng thông qua thư viện, firmware hay…

**\*\* Arduino IDE:**

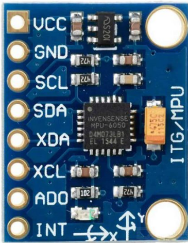
Arduino IDE là phần mềm mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Vì nó hỗ trợ cho Arduino nên để biên dịch mã vào các module / board khác thì cần phải cài đặt trong phần mềm.

**\*\* Unity:**

Unity là một game engine đa nền tảng, chủ yếu phát triển video game cho máy tính, console và điện thoại.

**SCHEMATICS**

Để có thể nạp code vào thiết bị nhúng cũng như kết nối với cảm biến gia tốc MPU6050, em đã lắp mạch như sau:

A picture containing text, different, colors

Description automatically generatedA picture containing text, electronics

Description automatically generated

**GPIO14**

**GPIO15**

**GND**

**3V3**

Diagram

Description automatically generated

**TRIỂN KHAI DỰ ÁN**

Dự án được triển khai theo tùng bước như sau:

2

3

4

5

1

Trong dự án môn học này, em sẽ trình bày các bước 2, 3, 4, 5.

1. Logo, company name

   Description automatically generated**VỀ EDGE IMPULSE**

Edge Impulse: [Edge Impulse](https://www.edgeimpulse.com/)

Edge Impulse là nền tảng cho phép xây dựng các dự án liên quan đến nghiên cứu máy móc trên vi điều khiển. Edge Impulse đưa ra tập hợp các API để giảm thiểu số lượng các bước thủ công cần thiết để có được dữ liệu bạn cần để đào tạo mô hình.

Hơn nữa, giao diện của Edge Impulse rất dễ sử dụng, trình tự các bước được sắp xếp theo thứ tự, bộ Document đầy đủ.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **THU THẬP DỮ LIỆU BẰNG EDGE IMPULSE**

Edge Impulse hỗ trợ người dùng rất nhiều cách để lấy data: nhận trực tiếp từ thiết bị nhúng, sử dụng điện thoại di động, sử dụng máy tính (dành cho nhận diện audio và hình ảnh), …

Ở đây, em sử dụng cách thu thập data từ điện thoại.

Graphical user interface, application, Teams

Description automatically generated

Khi chọn cách lấy data này, Edge Impulse sẽ cung cấp 1 mã QR code để quét và bắt đầu lấy data. Sau khi quét, màn hình máy tính sẽ hiển thị kết nối thành công và bắt đầu chọn loại data để thu thập. Ở đây em chọn thu thập data loại hành động (**Collecting motion**).

Text

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

Sau đó màn hình điện thoại sẽ chuyển đến trang lấy data. Nhập **label** cho dữ liệu, chiều dài dữ liệu. Ở đây em chọn phân loại là tự động tách thành 2 tập training và testing. Sau đó nhấn **Start recording** để bắt đầu thu dữ liệu.

Text, application

Description automatically generated

Dữ liệu hành động được thu rất đơn giản. Sau khi nhấn **Start recording**, chỉ cần lặp lại hành động mà mình muốn đưa vào mô hình học máy cho đến khi có được tỉ lệ 2 tập training, testing phù hợp (có thể theo dõi data thu được trong **Data acquisition**).

Rectangle

Description automatically generated

Tập dữ liệu được phân chia phù hợp là tập dữ liệu được chia thành 80% là tập training và 20% là tập testing. Nếu tỉ lệ 2 tập này của mỗi tập dữ liệu chưa được phân chia đồng đều thì thanh màu ở phần **Labels in your dataset** sẽ là màu đỏ.

1. **THIẾT KẾ IMPULSE**

Sau khi thu thập dữ liệu thì em sẽ tạo / thiết kế **Impulse**. Impulse cũng được hiểu như là xung dữ liệu, gồm 3 block chính:

1. **Input block:**

Input block cho biết loại dữ liệu đầu vào để xây dựng mô hình. Nó có thể là chuỗi thời gian (đối với dữ liệu âm thanh, hành động) hay dữ liệu hình ảnh.

Ở đây, vì dữ liệu em thu thập là hành động nên **Input block** sẽ là chuỗi thời gian (**Time series data**). Loại data này có những thông số như sau:

* **Input axes:** liệt kê các trục liên quan tới tập training.
* **Window size:** kích thước những đặc trưng thô được dùng để training.
* **Window increase:** tạo thêm đặc trưng “ảo” và chuyền cho **Learning block** thêm thông tin.
* **Frequency:** tự động tính toán dự trên mẫu training, có thể cập nhật lại giá trị này nhưng không thấp hơn 0.000016 (tương đương ít hơn 1 mẫu mỗi 60s).
* **Zero – pad data:** thêm giá trị 0 khi đặc trưng thô bị lỗi

Graphical user interface, application

Description automatically generated

1. **Processing block:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

Ấn vào khung như hình trên để tạo **Processing block. Processing block** bao gồm bộ xử lý tín hiệu số **DSP** (Digital Signal Processing) để tiến hành phân tách dữ liệu và xử lý.

Có nhiều loại **Processing block** để lựa chọn. Ở đây em chọn **Spectral Analysis**, tốt cho việc phân tích hành động. Dữ liêu hành động sẽ được trích ra tần số và đặc trưng của tín hiệu.

Graphical user interface, application, Teams

Description automatically generated Graphical user interface, application

Description automatically generated

1. **Learning block:**

**Learning block** là 1 neural network được train từ dữ liệu học. **Learning block** phụ thuộc vào mục đích xây dựng mô hình và loại dữ liệu đầu vào. Các thuật toán của **Learning block** bao gồm: classification, regression, anomaly detection, image transfer learning, keyword spotting, object detection. Ngoài ra, Edge Impulse còn hỗ trợ tự custom **Learning block**.

Ở đây em chọn thuật toán **classification**. Ý tưởng là mạng phân loại neural sẽ nhận 1 số đầu vào, và đầu ra là xác suất mà dữ liệu đầu vào đó thuộc về 1 lớp phân loại cụ thể.

Graphical user interface, application, Teams

Description automatically generated Graphical user interface, application

Description automatically generated

Sau khi tạo xong impulse, ta có được các đặc trưng như ý muốn dựa theo label của data. Nhấn **Save impulse** để lưu lại và đến bước tiếp theo.

A screenshot of a phone

Description automatically generated with medium confidence

1. **TRAINING MODEL**

Trong **Spectral features**, ta thấy được dữ liệu thô để xử lý tín hiệu. Các giá trị biến như sau:

Graphical user interface, application

Description automatically generated

Sau khi nhấn **Save parameters**, màn hình sẽ chuyển đến **Generate features**. Nhấn **Generate features** để tạo các đặc trưng. Trong mô hình này, em tạo ra 2 đặc trưng: left\_right và up\_down.

Graphical user interface, text, application, chat or text message

Description automatically generated Chart, scatter chart

Description automatically generated

Sau khi tạo ra các đặc trưng, em sẽ tiến hành train model. Các giá trị để train như sau:

Graphical user interface, application

Description automatically generated

Sau khi nhấn **Start training**, em thu được kết quả:

* Độ chính xác: 99.4%
* Sai số: 0.03

A picture containing graphical user interface

Description automatically generated Chart, scatter chart

Description automatically generated

1. **TEST MODEL**

Sau khi train model, em sẽ đến bước **test model** để kiểm tra mô hình học có tốt không. Khi nhấn vào **Model testing**, ta thấy được tập dữ liệu dùng để test.

**Graphical user interface

Description automatically generated with medium confidence**

Nhấn **Classify all**, dữ liệu sẽ được tiến hành test. Sau khi test xong, thu được kết quả: Độ chính xác là 98.1%. Điều đó chứng tỏ mô hình học tốt.

A picture containing timeline

Description automatically generated Chart, scatter chart, bubble chart

Description automatically generated

1. **TRIỂN KHAI MÔ HÌNH LÊN THIẾT BỊ NHÚNG**

Edge Impulse hỗ trợ triển khai mô hình lên thiết bị nhúng theo nhiều cách: tạo thư viện .zip, tạo firmware, chạy trực tiếp. Trong dự án này, em triển khai bằng cách tạo thư viện .zip. Do chương trình mã được tạo trên Arduino IDE và sau đó biên dịch vào thiết bị nhúng, em chọn create **Arduino library**.

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application, chat or text message

Description automatically generated

Sau khi nhấn **Build**, 1 file .zip sẽ được tạo ra và tự động tải về máy.

1. **FILE PROJECT ARDUINO**

Để có thể biên dịch mã vào ESP32 – CAM, em cài đặt package ESP32 trên Arduino. Để cài được package này, trước hết mình phải điền link **Preferences**:

Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Link preferences này có thể tìm thấy trên Github. Sau khi nhấn **OK**, em vào **Boards Manager** và tìm **ESP32**. Sau khi cài đặt package này, em tiếp tục vào **Library Manager** cài đặt thư viện **Adafruit\_MPU6050**. Đây là thư viện hỗ trợ lập trình đối với cảm biến gia tốc MPU6050.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Để sử dụng được thư viện **Adafruit** cũng như mô hình vừa train trên Edge Impulse, ta phải include thư viện vào file project:

**Text

Description automatically generated**

Trước tiên, ta định nghĩa các giá trị ban đầu và khai báo các biến sử dụng xuyên suốt chương trình:

Graphical user interface, text

Description automatically generated Graphical user interface, text, application

Description automatically generated

Như hình trên, em có định nghĩa 2 giá trị: **SDA = 14** và **SCL = 15**. Đây là 2 đầu đọc tín hiệu của cảm biến MPU6050. 2 giá trị này có nghĩa là đầu **SDA** của MPU6050 sẽ nối với chân **GPIO14** của ESP32 – CAM, còn đầu **SCL** của MPU6050 sẽ nối với chân **GPIO15** của ESP32 \_ CAM. Để kết nối thành công, ta phải thêm câu lệnh vào hàm **void setup()**:



Trong đoạn mã chương trình, em có viết thêm hàm **find\_Max()**. Tham số truyền vào là kết quả sau khi test input và đầu ra là **label** cho xác suất lớn nhất.

Graphical user interface, text

Description automatically generated

Vì tính chất việc điều khiển game chỉ có thể là left hoặc right, up hoặc down, do đó nếu chương trình trả về chuỗi như label là left\_right hoặc up\_down thì game không thể nhận hành động chính xác được. Vì vậy, sau khi test input và tìm ra label của hành động đó, em tiến hành phân loại hành động.

Polygon

Description automatically generated Diagram, polygon

Description automatically generated

Tìm label có xác suất lớn nhất trong số các label được test: **max**

**\*\* Nếu max >= 0.6:**

* Nếu **label == “left\_right”**:
  + Nếu **trục x đầu vào < 0**: left
  + Nếu **trục x đầu vào > 0**: right
  + Nếu **trục x đầu vào == 0**: nothing
* Nếu **label == “up\_down”**:
  + Nếu **trục y đầu vào > 0**: up
  + Nếu **trục y đầu vào < 0**: down
  + Nếu **trục y đầu vào == 0**: nothing

**\*\* Nếu max < 0.6:**

* Đặt cờ **flag = 0**
* Nếu xác suất của các label còn lại cách **max** 1 khoảng <= **epsilon** (epsilon khai báo ban đầu là 0.1), gây nhiễu output: nothing, gán lại **flag = -1**
* Nếu không gây nhiễu (**flag** vẫn bằng 0) thì tiếp tục xét label và đầu vào trục x / trục y như trường hợp trên.

1. **KẾT NỐI ARDUINO VỚI UNITY**

Để có thể kết nối Arduino với Unity, em sử dụng Ardity. Ardity là 1 free plugin hỗ trợ kết nối Arduino IDE với Unity thông qua port COM.

Graphical user interface, application

Description automatically generated

Để sử dụng được Ardity, ta phải tải Ardity từ Unity **Asset Store**, sau đó **import** package vào project game muốn kết nối.

Graphical user interface, text, application

Description automatically generated

Sau khi import xong, ta vào **Assets / Ardity / Scripts** và mở file **SerialController** cập nhật **portName** và **baudRate**. **portName** là port Arduino kết nối với thiết bị nhúng và **baudRate** được lấy từ Arduino IDE.

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, application

Description automatically generated

Sau khi cập nhật xong **portName** và **baudRate**, ta vào **Assets / Scripts / <Object>** để tạo file **ReadInput**. File này có nhiệm vụ đọc input từ thiết bị nhúng thông qua **portName** và xử lý hành động mong muốn. Sau đó, vào **Prefabs / <Object>**, add component **ReadInput** cho <Object>. Trong file **ReadInput** và file script của <Object> cũng phải khai báo biến để sử dụng component.

Text

Description automatically generated**\*\* Khai báo và xây dựng hàm trong file ReadInput:**





Hàm **Movement()** dùng để di chuyển trên game dựa theo đầu vào nhận được.

A screenshot of a computer screen

Description automatically generated with medium confidence

**ĐÁNH GIÁ TỔNG QUÁT**

1. **Dữ liệu thu được:**

Trong quá trình thu thập dữ liệu, em nhận thấy một số dữ liệu đầu vào dù khác label nhưng raw data vẫn giống nhau, điều đó khiến cho việc học máy trở nên khó khăn hơn 1 tí, dễ gây nhiễu và khi test trên thiết bị gây ra 1 số sai sót.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

1. **Kết quả sau khi train model:**

Sau khi train model, kết quả cho thấy sai số nhỏ, độ chính xác cao, chứng tỏ mô hình học tốt.

1. **Kết quả sau khi test model:**

Sau khi test model, kết quả cho thấy độ chính xác cao, chứng tỏ mô hình này là đúng đắn.

1. **Đưa mô hình vào thiết bị nhúng và điều khiển:**

Do sự nhiễu data từ khi thu thập nên việc nhận diện hành động thông qua thiết bị vẫn còn sai sót. Bên cạnh đó, khi điều khiển với Unity, chương trình còn chậm và lag.

**DEMO ĐIỀU KHIỂN THIẾT BỊ TRÊN UNITY**

Link video demo thiết bị xuất kết quả ra màn hình Serial Monitor trên Arduino IDE: [(92) DEMO CHẠY THIẾT BỊ NHÚNG VÀ XUẤT KẾT QUẢ RA MÀN HÌNH SERIAL MONITOR - YouTube](https://www.youtube.com/watch?v=EpvXc3buXqc)

Link video demo điều khiển thiết bị trên game Unity: [(92) DEMO ĐIỀU KHIỂN THIẾT BỊ NHÚNG TRÊN GAME UNITY - YouTube](https://www.youtube.com/watch?v=TGZN0gYCTP4)