

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HUNG YÊN**

**ĐẶNG NGỌC NHÂN**

**THIẾT KẾ ROBO Ô TÔ ĐIỀU KHIỂN TỪ XA**

**ĐỒ ÁN 4**

**HUNG YÊN - 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HÙNG YÊN**

**ĐẶNG NGỌC NHÂN**

**THIẾT KẾ ROBO Ô TÔ ĐIỀU KHIỂN TỪ XA**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**  
**CHUYÊN NGÀNH: PHÁT TRIỂN ỨNG DỤNG IOT**

**ĐỒ ÁN 4**

**NGƯỜI HƯỚNG DẪN**  
**TRỊNH VĂN LOAN**

**HÙNG YÊN - 2023**

Nhận xét của giảng viên 1 đánh giá quá trình:

.....

.....

.....

.....

.....

Ký và ghi họ tên

Nhận xét của giảng viên 2 đánh giá quá trình:

.....

.....

.....

.....

.....

.....

.....

Ký và ghi họ tên

Nhận xét của giảng viên hướng dẫn:

.....

.....

.....

.....

.....

.....

.....

.....

Ký và ghi họ tên

## LỜI CAM ĐOAN

Tôi xin cam đoan đề án “Thiết kế robo ô tô điều khiển từ xa” là công trình nghiên cứu của bản thân. Những nội dung sử dụng trong đề án không sao chép của bất cứ tài liệu nào. Những nội dung trích dẫn được thực hiện đúng theo quy định về vi phạm bản quyền. Các kết quả trình bày trong đề án hoàn toàn là kết quả do bản thân tôi và các thành viên trong nhóm thực hiện, nếu sai tôi xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng yên, ngày ... tháng ... năm.....*

Sinh viên

.....

## LỜI CẢM ƠN

.....

.....

.....

.....

.....

.....

.....

.....

.....

## MỤC LỤC

DANH SÁCH TỪ VIẾT TẮT .....	8
DANH SÁCH HÌNH VẼ .....	9
DANH SÁCH BẢNG BIỂU .....	11
CHƯƠNG 1: MỞ ĐẦU .....	12
1.1. Lý do chọn đề tài .....	12
1.2. Mục tiêu của đề tài .....	13
1.2.1. Mục tiêu tổng quát.....	13
1.2.2. Mục tiêu cụ thể .....	13
1.3. Giới hạn và phạm vi của đề tài .....	14
1.3.1. Đối tượng nghiên cứu.....	14
1.3.2. Phạm vi nghiên cứu .....	14
1.4. Nội dung thực hiện .....	14
1.5. Phương pháp tiếp cận .....	15
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	16
2.1. Ngoại vi .....	16
2.1.1. Vi điều khiển arduino uno R3 .....	16
2.1.2. Modul bluetooth HC05.....	21
2.1.3. Động cơ servo (Động cơ servo SG90) .....	24
2.1.4. Motor DC 5V.....	25
2.1.5. Mô hình ô tô .....	27
2.1.6. Cánh tay robot .....	28
2.2. Phần mềm cho hệ thống .....	29

2.2.1. Phần mềm Proteus .....	29
2.2.2. Phần mềm Arduino IDE .....	31
2.2.3. Android Studio .....	35
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	37
3.1. Đặc tả yêu cầu hệ thống .....	37
3.1.1. Các yêu cầu chức năng .....	37
3.1.2. Các yêu phi cầu chức năng .....	37
3.2. Thiết kế hệ thống .....	38
3.2.1. Thiết kế phần cứng cho hệ thống .....	38
3.2.2. Thiết kế phần mềm cho hệ thống .....	42
CHƯƠNG 4: TÍCH HỢP VÀ ĐÁNH GIÁ HỆ THỐNG .....	44
4.1. Xây dựng và tích hợp hệ thống .....	44
4.1.1. Xây dựng hệ thống .....	44
4.1.2. Tích hợp giữa phần cứng hệ thống và phần mềm .....	68
4.2. Kiểm thử và đánh giá hệ thống .....	70
4.3. Hướng dẫn vận hành hệ thống .....	72
KẾT LUẬN .....	74
TÀI LIỆU THAM KHẢO .....	75

## **DANH SÁCH TỪ VIẾT TẮT**

<b>Từ viết tắt</b>	<b>Nghĩa tiếng Anh</b>	<b>Nghĩa tiếng Việt</b>
IoT	Internet of Things	Internet kết nối vạn vật
LCD	Liquid-Crystal Display	Màn hình tinh thể lỏng
LED	Light-Emitting Diode	Diode phát quang
PIC	Programmable Intelligent Computer	Máy tính thông minh khả trình



## DANH SÁCH HÌNH VẼ

Hình 2. 1: Vi điều khiển arduino uno R3 .....	16
Hình 2. 2: Các chân vào ra của arduino uno r3 .....	19
Hình 2. 3: Arduino IDE (Intergrated Development Environment) .....	21
Hình 2. 4: Modul bluetooth HC05.....	22
Hình 2. 5: Động cơ servo SG90 .....	25
Hình 2. 6: Motor DC 5V .....	26
Hình 2. 7: Mô hình ô tô 4 bánh .....	27
Hình 2. 8: Cánh tay robot .....	28
Hình 2. 9: Phần mềm proteus .....	29
Hình 2. 10: Arduino IDE.....	32
Hình 2. 11: Android studio.....	36
Hình 3. 1: Sơ đồ khối .....	38
Hình 3. 2: Sơ đồ nguyên lý.....	39
Hình 3. 3: Khối điều khiển trung tâm.....	40
Hình 3. 4: Khối điều khiển động cơ motor.....	41
Hình 3. 5: Khối điều khiển cánh tay robot .....	41
Hình 3. 6: Khối điều khiển bluetooth .....	42
Hình 3. 7: Điều khiển xe .....	42
Hình 3. 8: Điều khiển cánh tay robot .....	43
Hình 3. 9: Phần cứng hệ thống .....	44
Hình 3. 10: Lập trình app android điều khiển .....	49
Hình 3. 11: Giao diện app điều khiển.....	67

Hình 3. 12: Kết nối arduino với máy tính .....	68
Hình 4. 1: Vận hành hệ thống.....	72

## **DANH SÁCH BẢNG BIỂU**

## CHƯƠNG 1: MỞ ĐẦU

### 1.1. Lý do chọn đề tài

Điều khiển ô tô cánh tay robot từ xa là một trong những đề tài công nghệ thu hút được sự quan tâm của nhiều nhà nghiên cứu và sinh viên. Trong báo cáo này, tôi sẽ trình bày lý do chọn đề tài này và giải thích tầm quan trọng của nó.

Đề tài của tôi là “Thiết kế robo ô tô điều khiển từ xa”. Với đề tài này, tôi sẽ thiết kế ra một mô hình ô tô có cánh tay robot mọi thứ đều được điều khiển từ xa:

- Ứng dụng thực tiễn: Việc điều khiển ô tô cánh tay robot từ xa có thể được áp dụng trong nhiều lĩnh vực thực tiễn, như là việc điều khiển các robot trong sản xuất công nghiệp, trong ngành y tế, giám sát tại các khu vực nguy hiểm, hoặc trong các hoạt động khảo sát địa hình, tìm kiếm cứu nạn.

- Tiết kiệm chi phí: Sử dụng Arduino để điều khiển ô tô cánh tay robot từ xa là một giải pháp có chi phí thấp và hiệu quả, giúp tiết kiệm chi phí đầu tư và thời gian phát triển.

- Linh hoạt và dễ dàng mở rộng: Arduino cho phép người dùng dễ dàng mở rộng và tùy chỉnh các chức năng, kết nối với các cảm biến và thiết bị khác nhau để điều khiển ô tô cánh tay robot từ xa theo nhu cầu và mục đích sử dụng.

- Tiềm năng ứng dụng trong tương lai: Việc phát triển công nghệ điều khiển ô tô cánh tay robot từ xa bằng Arduino có thể giúp cho việc phát triển các công nghệ liên quan đến các thiết bị tự động hóa, robot và các hệ thống IoT trong tương lai.

Tầm quan trọng của đề tài:

- Giúp nâng cao hiệu suất công việc: Việc điều khiển ô tô cánh tay robot từ xa giúp nâng cao hiệu suất công việc và giảm tải nhân lực trong các công việc khó khăn và nguy hiểm.

- Tiết kiệm chi phí: Việc sử dụng robot để thực hiện các công việc khó khăn và nguy hiểm sẽ giúp giảm tải nhân lực và tiết kiệm chi phí đáng kể trong các ngành công nghiệp và sản xuất.

- Cải thiện an toàn và hiệu quả: Việc sử dụng ô tô cánh tay robot để thực hiện các tác vụ nguy hiểm có thể giảm thiểu nguy cơ tai nạn lao động.

- Đóng góp vào sự phát triển của công nghệ: Nghiên cứu và phát triển các hệ thống điều khiển ô tô cánh tay robot từ xa bằng Arduino có thể đóng góp vào sự phát triển của các công nghệ liên quan đến robot và hệ thống IoT trong tương lai.

Đề tài này có tính ứng dụng cao, tiết kiệm chi phí, linh hoạt và dễ dàng mở rộng, và có tiềm năng ứng dụng trong tương lai. Việc phát triển và nghiên cứu các hệ thống điều khiển ô tô cánh tay robot từ xa là rất cần thiết để nâng cao hiệu suất công việc, giảm thiểu nguy cơ tai nạn lao động và đóng góp vào sự phát triển của công nghệ.

## **1.2. Mục tiêu của đề tài**

### ***1.2.1. Mục tiêu tổng quát***

Xây dựng một hệ thống điều khiển robot ô tô cánh tay từ xa, giúp tăng hiệu quả sản xuất, giảm thiểu nguy cơ tai nạn lao động và nâng cao độ an toàn trong môi trường làm việc. Đồng thời, đề tài còn nhằm mục đích nghiên cứu và phát triển các công nghệ liên quan đến robot và hệ thống IoT, đóng góp vào sự phát triển của công nghệ trong tương lai. Hệ thống điều khiển ô tô cánh tay robot từ xa được thiết kế dựa trên nền tảng Arduino, với khả năng linh hoạt, mở rộng và dễ dàng sử dụng.

### ***1.2.2. Mục tiêu cụ thể***

- Tìm hiểu và nghiên cứu về cấu tạo, nguyên lý hoạt động và tính năng của robot ô tô cánh tay.
- Tìm hiểu và phát triển các công nghệ liên quan đến robot và hệ thống IoT, bao gồm các loại cảm biến, vi điều khiển, kết nối mạng và giao thức truyền thông.
- Thiết kế và xây dựng hệ thống điều khiển ô tô cánh tay robot từ xa bằng Arduino, với khả năng điều khiển và giám sát các chức năng của robot như di chuyển, nâng hạ, xoay tay, v.v.
- Kiểm tra và đánh giá hiệu suất của hệ thống điều khiển ô tô cánh tay robot từ xa bằng Arduino, bao gồm độ chính xác, độ tin cậy và độ ổn định.

- Nghiên cứu và đề xuất các cải tiến và phát triển cho hệ thống điều khiển ô tô cánh tay robot từ xa bằng Arduino, nhằm tăng cường tính năng và hiệu suất của hệ thống.

### **1.3. Giới hạn và phạm vi của đề tài**

#### ***1.3.1. Đối tượng nghiên cứu***

- Tìm hiểu và nghiên cứu về cấu tạo, nguyên lý hoạt động và tính năng của robot ô tô cánh tay.
- Tìm hiểu và phát triển các công nghệ liên quan đến robot và hệ thống IoT, bao gồm các loại cảm biến, vi điều khiển, kết nối mạng và giao thức truyền thông.
- Đề tài hướng đến các doanh nghiệp và tổ chức trong lĩnh vực sản xuất và công nghiệp, những nơi có nhu cầu sử dụng robot ô tô cánh tay để tăng cường năng suất, giảm thiểu nguy cơ tai nạn lao động và nâng cao độ an toàn trong môi trường làm việc.

#### ***1.3.2. Phạm vi nghiên cứu***

Nghiên cứu và phát triển các kỹ thuật điều khiển và xử lý tín hiệu mới nhằm tăng cường tính năng và hiệu suất của hệ thống, ví dụ như kỹ thuật điều khiển bằng giọng nói, điều khiển qua app được lập trình trên thiết bị smartphone ...

Nghiên cứu và áp dụng các công nghệ mới như truyền thông không dây vào hệ thống điều khiển ô tô cánh tay robot từ xa để nâng cao tính bảo mật và độ tin cậy của hệ thống.

Nghiên cứu và phát triển các giải pháp điều khiển cho ô tô cánh tay robot để đáp ứng nhu cầu của các ứng dụng khác nhau.

Nghiên cứu và đề xuất các giải pháp tối ưu hóa hiệu suất và tiết kiệm năng lượng cho hệ thống điều khiển ô tô cánh tay robot từ xa bằng cách sử dụng các kỹ thuật tối ưu hóa tiêu thụ điện năng và quản lý năng lượng.

### **1.4. Nội dung thực hiện**

- Chuẩn bị một số các linh kiện :

- + Mô hình xe ô tô
- + Vi điều khiển arduino, node mcu ch340
- + Mô hình cánh tay robot, servo
- + Modul HC05 điều khiển qua bluetooth
- + Lập trình app android để điều khiển từ xa.

### **1.5. Phương pháp tiếp cận**

- Tiến hành thu thập và phân tích những thông tin, tài liệu liên quan đến đề tài
- Xác định các yêu cầu phân tích thiết kế hệ thống chương trình cho phù hợp.
- Xây dựng chương trình theo những yêu cầu đã đặt ra.
- Triển khai chương trình và đánh giá kết quả đạt được.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Ngoại vi

#### 2.1.1. Vi điều khiển arduino uno R3



**Hình 2. 1:** Vi điều khiển arduino uno R3

Arduino Uno R3 được sử dụng vi điều khiển ATmega328, tương thích với hầu hết các loại Arduino Shield trên thị trường, có thể gắn thêm các module mở rộng để thực hiện thêm các chức năng như điều khiển motor, kết nối wifi hay các chức năng khác.

Sử dụng ngôn ngữ lập trình C,C++ hoặc Arudino, một ngôn ngữ bắt nguồn từ C,C++ trên phần mềm riêng cho lập trình Arduino IDE.

#### ❖ Các chân năng lượng

Arduino Uno R3 được cấp nguồn 5V qua cáp usb hoặc cấp nguồn ngoài thông qua Adaptor chuyển đổi , với điện áp khuyến dùng là khoảng 6-9V. Có thể cấp nguồn từ máy tính qua cổng USB.

Các chân 5V, 3.3V là chân dùng để cấp nguồn đầu ra cho các thiết bị, không phải chân cấp nguồn vào.



- **Vin(Voltage Input):** Dùng để cấp nguồn ngoài cho Arduino Uno, nối dương cực vào chân nà và cực âm vào chân GND.
- **GND(Ground):** Cực âm của nguồn điện cấp cho Arduino Uno. Khi sử dụng các thiết bị sử dụng những nguồn điện riêng biệt thì phải nối các chân này.
- **IOREF:** Điện áp hoạt động của Arduino, có mức điện áp là 5V. Không được sử dụng để lấy nguồn từ chân này.
- **RESET:** Việc nhấn nút RESET trên mạch arduino tương tự như khi nối chân RESET với GND qua điện trở 10KΩ.

❖ **Một vài thông số kỹ thuật của arduino uno r3**

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)

EEPROM	1 KB (ATmega328)
--------	------------------

❖ **Lưu ý:**

- Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO.
- Việc làm chập mạch nguồn vào của Arduino UNO sẽ khiến hỏng mạch có thể gây chập cháy.
- Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích.
- Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.
- Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.
- Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.
- Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.
- Cường độ dòng điện qua một chân Digital hoặc Analog bất kỳ của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

❖ **Bộ nhớ**

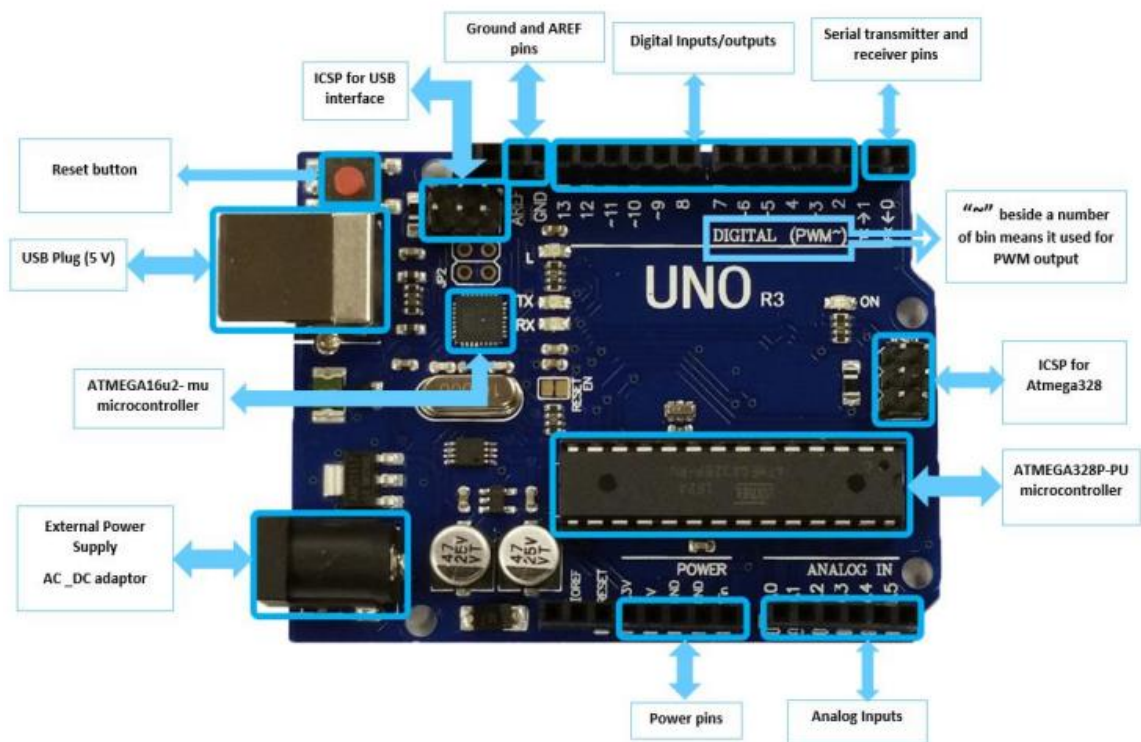
Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

- **32KB bộ nhớ Flash:** những đoạn lệnh lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
- **2KB cho SRAM (Static Random Access Memory):** giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Khai báo càng nhiều biến thì càng cần nhiều bộ nhớ

RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

- **1KB EEPROM** (Electrically Eraseable Programmable Read Only Memory): đây giống như một chiếc ổ cứng mini – nơi có thể đọc và ghi dữ liệu vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

#### ❖ Các cổng ra vào



**Hình 2. 2:** Các chân vào ra của arduino uno r3

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

- **2 chân Serial:** 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối

Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết

- **Chân PWM (~): 3, 5, 6, 9, 10, và 11:** cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ  $0 \rightarrow 2^8-1$  tương ứng với  $0V \rightarrow 5V$ ) bằng hàm `analogWrite()`. Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

- **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

- **LED 13:** trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

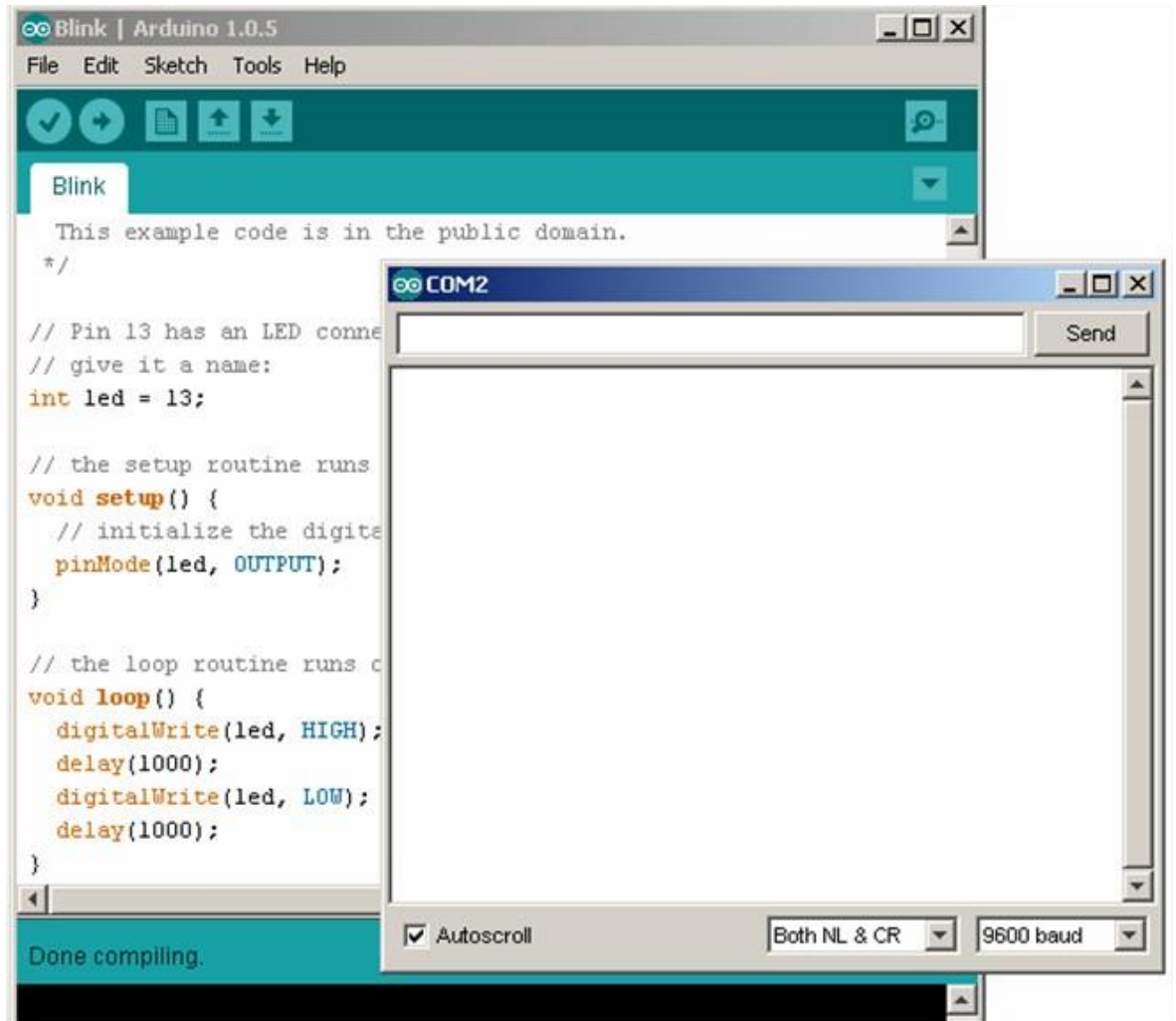
- Arduino UNO có 6 chân analog ( $A0 \rightarrow A5$ ) cung cấp độ phân giải tín hiệu 10bit ( $0 \rightarrow 2^{10}-1$ ) để đọc giá trị điện áp trong khoảng  $0V \rightarrow 5V$ . Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ  $0V \rightarrow 2.5V$  với độ phân giải vẫn là 10bit.

- Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

### ❖ Lập trình cho Arduino

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “*ngôn ngữ Arduino*”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ thở đối với bạn.

Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cung cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (Integrated Development Environment) như hình dưới đây.



**Hình 2. 3:** *Arduino IDE (Intergrated Development Environment)*

### **2.1.2. Modum bluetooth HC05**

Module thu phát Bluetooth HC-05 dùng để thiết lập kết nối Serial giữa 2 thiết bị bằng sóng bluetooth. Điểm đặc biệt của module bluetooth HC-05 là module có thể hoạt động được ở 2 chế độ: MASTER hoặc SLAVE.

+ Ở chế độ SLAVE: bạn cần thiết lập kết nối từ smartphone, laptop, usb bluetooth để dò tìm module sau đó pair với mã PIN là 1234. Sau khi pair thành công, bạn đã có 1 cổng serial từ xa hoạt động ở baud rate 9600.

+ Ở chế độ MASTER: module sẽ tự động dò tìm thiết bị bluetooth khác (1 module bluetooth HC-06, usb bluetooth, bluetooth của laptop...) và tiến hành pair chủ động mà không cần thiết lập gì từ máy tính hoặc smartphone.

Module Bluetooth thu phát HC-05 được thiết kế nhỏ gọn ra chân tín hiệu giao tiếp cơ bản và nút bấm để vào chế độ AT COMMAND, mạch được thiết kế để có thể cấp nguồn và giao tiếp qua 3.3VDC hoặc 5VDC, thích hợp cho nhiều ứng dụng khác nhau: Robot Bluetooth, điều khiển thiết bị qua Bluetooth,....



*Hình 2. 4: Modul bluetooth HC05*

❖ **Tổng quan về module HC-05**

- HC05 tuân theo giao thức “Bluetooth V2.0 + EDR” (EDR là viết tắt của tốc độ dữ liệu cao)
- Tần số hoạt động của nó là băng tần ISM 2,4 GHz
- HC05 sử dụng hệ thống Bluetooth CSR Bluecore 04-chip đơn bên ngoài với công nghệ CMOS
- Module này tuân theo giao thức chuẩn 802.15.1 của IEEE (Institute of Electrical and Electronics Engineers)
- Module này sử dụng (FHSS), một kỹ thuật mà tín hiệu vô tuyến được gửi ở các mức tần số khác nhau

❖ **THÔNG SỐ MODULE THU PHÁT BLUETOOTH HC-05**

- Điện áp hoạt động: 3.3 ~ 5VDC

- Mức điện áp chân giao tiếp: TTL tương thích 3.3VDC và 5VDC.
- Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA.
- Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Support profiles: Bluetooth serial port (master and slave)
- Bluetooth protocol: Bluetooth specification v2.0 + EDR
- Frequency: 2.4 GHz ISM band
- Modulation: GFSK (Gaussian frequency shift keying)
- Transmit power: =4 dBm, class 2
- Sensitivity: =-84 dBm at 0.1% BER
- Rate: Asynchronous: 2.1 Mbps (max.)/160 kbps
- Synchronous: 1 Mbps/1 Mbps
- Security features: authentication and encryption
- Kích thước: 15.2 x 35.7 x 5.6mm

❖ **THIẾT LẬP MẶC ĐỊNH:**

- Thiết lập UART mặc định: Baudrate 9600, N, 8, 1.
- Pairing code mặc định: 1234 hoặc 0000.
- Để vào chế độ AT COMMAND, bấm và giữ nút trước khi cấp nguồn, LED sẽ nháy 2s. baud rate cho chế độ AT COMMAND là 38400. Chân Tx nối với chân Rx. Lưu ý các lệnh AT đều là chữ in hoa.
- Cấp nguồn và không nhấn nút sẽ chạy bình thường. LED sẽ nháy nhanh
- Chân EN chỉ nhận mức logic TTL 3V3. Không có chức năng chọn vào chế độ AT COMMAND.

#### PIN OUT HC-05

Pin#1	Enable Pin	Đặt giá trị HIGH hoặc LOW
Pin#2	Vcc	Chân cấp nguồn
Pin#3	GND	Ground
Pin#4	TX	Chân truyền
Pin#5	RX	Chân nhận
Pin#6	State	Có thể kết nối với Led ngoài

#### 2.1.3. Động cơ servo (Động cơ servo SG90)

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0 độ - 180 độ. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô hình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và không sắc chắc chắn,...

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.



Động Cơ Servo SG90 là loại động cơ được dùng phổ biến trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.



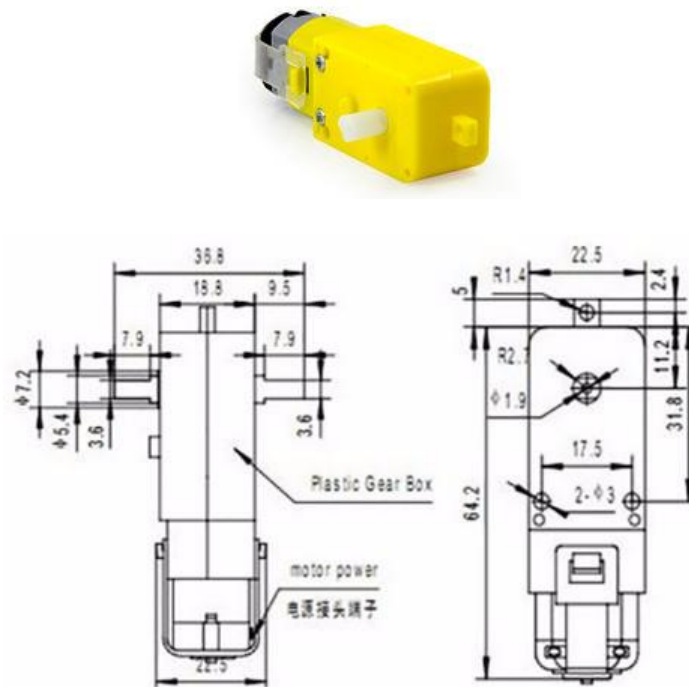
*Hình 2. 5: Động cơ servo SG90*

❖ **Thông Số Kỹ Thuật Động Cơ Servo**

- SG90 Khối lượng : 9g
- Kích thước: 23mmX12.2mmX29mm
- Momen xoắn: 1.8kg/cm Tốc độ hoạt động: 60 độ trong 0.1 giây
- Điện áp hoạt động: 4.8V(~5V)
- Nhiệt độ hoạt động: 0 °C – 55 °C
- Động Cơ Servo SG90 góc quay 180 độ

**2.1.4. Motor DC 5V**

Tốc độ cao lên đến 200 vòng/phút, tại 200mA/6V và 90 vòng/phút tại 150mA/3V.

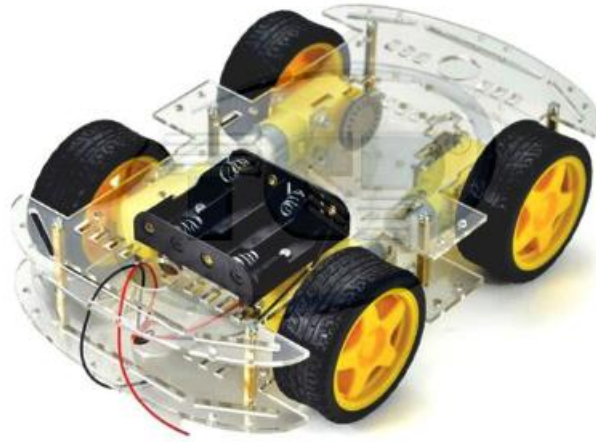


*Hình 2. 6: Motor DC 5V*

❖ **Thông số kỹ thuật**

- Motor: Động Cơ Cốt Lõi
- Tốc Độ vận hành (V 3): 90 +/-10% RPM
- Tốc Độ vận hành (V 6): 200 +/-10% RPM
- Stall Torque (6 V): 0.8kg.cm
- Kích thước: 70x22x18 mét
- Trọng lượng: 30 gam
- Tỷ số Truyền: 1:48 DC130 TT
- Hộp động cơ GearRatio: 1: 48, 1: 120, 1: 180, 1:220, 1:256, 1: 288) tổng 6 loại
- Điện Áp hoạt động: 3.0 ~ 12.0 V
- Đầu ra Mô-men Xoắn: 1 ~ 5 kgf. cm
- ĐỘNG CƠ: carbon Dây nối Dài: 100 mét.

### 2.1.5. Mô hình ô tô



**Hình 2. 7:** Mô hình ô tô 4 bánh

#### ❖ Thông số kỹ thuật

Bộ sản phẩm bao gồm:

- 2 đế nhựa mica trong suốt
- 4 động cơ hộp giảm tốc, điện áp cung cấp cho động cơ 6-9V
- 4 bánh xe robot bằng nhựa , lớp chất liệu cao su
- 1 bộ ốc vít và ke để gắn động cơ vào khung xe
- 1 bộ cọc đồng để ghép 2 tấm đế mica

Kích thước của khung xe : chiều dài khoảng 25cm, chiều rộng 14,8 cm, đường kính bánh xe 6cm, khung xe robot có những ưu điểm sau:

- Cơ cấu cơ khí đơn giản, dễ dàng lắp ghép
- Sử dụng 4 động cơ giảm tốc DC chuyên hướng linh hoạt dựa trên xử lý tốc độ của động cơ, Khung xe có kích thước lớn, rất dễ để mở rộng, lắp đặt các module, mạch điện hỗ trợ.

#### ❖ Tính năng ứng dụng

- Dùng dễ dàng lắp ghép các động cơ lên các tám phẳng.

- Chế tạo xe điều khiển, Robot rò đường line...vv
- Tốt hơn loại khung xe robot 3 bánh, dễ dàng điều chỉnh hướng đi của xe, xoay xe dựa trên việc điều khiển tốc độ của bánh xe thích hợp cho những ứng dụng làm xe robot tự hành, tránh vật cản, lắp ghép với cánh tay robot để điều khiển.

#### **2.1.6. Cánh tay robot**

Cánh tay robot là một phần của robot được sử dụng để thực hiện các nhiệm vụ như di chuyển, vận chuyển và định vị các đối tượng trong không gian ba chiều. Nó được xây dựng từ nhiều khớp nối và các bộ phận có thể di chuyển như động cơ, bánh răng và cảm biến.

Cánh tay robot có thể được điều khiển bằng các phương pháp khác nhau, bao gồm điều khiển thủ công thông qua bộ điều khiển hoặc tự động hóa thông qua phần mềm điều khiển robot. Với các công nghệ như máy móc học và trí tuệ nhân tạo, các cánh tay robot ngày nay đã trở nên rất linh hoạt và có thể thực hiện các nhiệm vụ phức tạp trong nhiều lĩnh vực, bao gồm sản xuất, y tế, khoa học và quân sự.

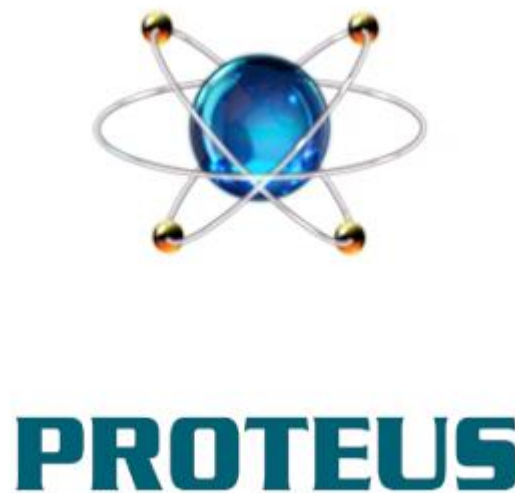


**Hình 2. 8:** Cánh tay robot

## 2.2. Phần mềm cho hệ thống

### 2.2.1. Phần mềm Proteus

Phần mềm vẽ Proteus là phần mềm vẽ mạch điện tử được phát triển bởi công ty Lancer Electronics. Phần mềm có thể mô tả hầu hết các Linh Kiện Điện Tử thông dụng hiện nay, đặc biệt hỗ trợ cho cả các phần mềm như 8051, PIC, Motorola, AVR.



**Hình 2. 9:** Phần mềm proteus

Proteus có khả năng mô phỏng hoạt động của các mạch điện tử bao gồm phần thiết kế mạch và viết trình điều khiển cho các loại vi điều khiển như MCS-51, AVR, PIC...

Có 2 chương trình trong phần mềm đó là ARES dùng trong vẽ mạch in và ISIS sử dụng cho mô phỏng mạch. Trong 2 chương trình này thì ISIS có phần nổi bật hơn so với ARES. ISIS đã được phát triển trong 12 năm và có tới hơn 12000 người dùng trên khắp thế giới (chắc chắn con số hiện tại đã tăng hơn rất nhiều. Điểm nổi bật của chúng đó là khả năng mô phỏng hoạt động của các vi điều khiển mà không cần dùng thêm bất kỳ một phần mềm phụ trợ nào khác. Từ phần mềm ISIS có thể dễ dàng chuyển sang ARES hoặc bất kỳ phần mềm vẽ mạch in khác.

Hình ảnh mạch điện được tạo bởi ISIS rất đẹp và dễ nhìn, chúng cho phép ta tùy chọn các đường nét, các màu sắc mạch điện hoặc các thiết kế theo các templates.

Ngoài ra phần mềm mô phỏng mạch của Proteus có khả năng sắp xếp các đường mạch và vẽ điểm giao mạch tự động.

Những đặc điểm nổi bật của proteus:

- Có khả năng mô phỏng hầu hết trình điều khiển cho vi điều khiển
- Chọn đối tượng và thiết lập thông số cho đối tượng dễ dàng
- Xuất ra file Netlist tương thích với các chương trình làm mạch in thông dụng.
- Xuất file thông kê linh kiện cho mạch.
- ISIS tích hợp nhiều công cụ giúp cho việc quản lý mạch điện lớn, mạch điện có thể lên đến hàng ngàn linh kiện phục vụ cho thiết kế mạch chuyên nghiệp.
- Thiết kế theo cấu trúc (hierarchical design).
- Khả năng tự động đánh số linh kiện.

### **Cách tạo mạch trong Proteus**

Bắt đầu thiết kế mới

- Bước 1: Mở phần mềm ISIS và chọn New design trong menu File.
- Bước 2: Một hộp thoại xuất hiện để lưu thiết kế hiện tại. Tuy nhiên, chúng ta đang tạo một file thiết kế mới nên bạn có thể nhấp vào Yes hoặc No tùy thuộc vào nội dung của file hiện tại. Sau đó, một cửa sổ bật lên xuất hiện yêu cầu chọn mẫu. Việc chọn khổ giấy trong khi in cũng tương tự như vậy. Bây giờ hãy chọn mặc định hoặc theo kích thước layout của mạch.
- Bước 3: Một sheet thiết kế không có tiêu đề sẽ được mở ra, hãy lưu nó theo ý muốn của bạn, tốt hơn hết là bạn nên tạo một thư mục mới cho mọi layout vì nó tạo ra các tệp khác hỗ trợ thiết kế của bạn. Tuy nhiên, cái này không phải là bắt buộc.
- Bước 4: Để chọn linh kiện, nhấp vào nút Component Mode.
- Bước 5: Nhấp vào Pick from Libraries. Nó hiển thị các danh mục của các linh kiện có sẵn và một tùy chọn tìm kiếm để nhập tên linh kiện.
- Bước 6: Chọn các linh kiện từ danh mục hoặc nhập tên linh kiện vào ô Keywords.

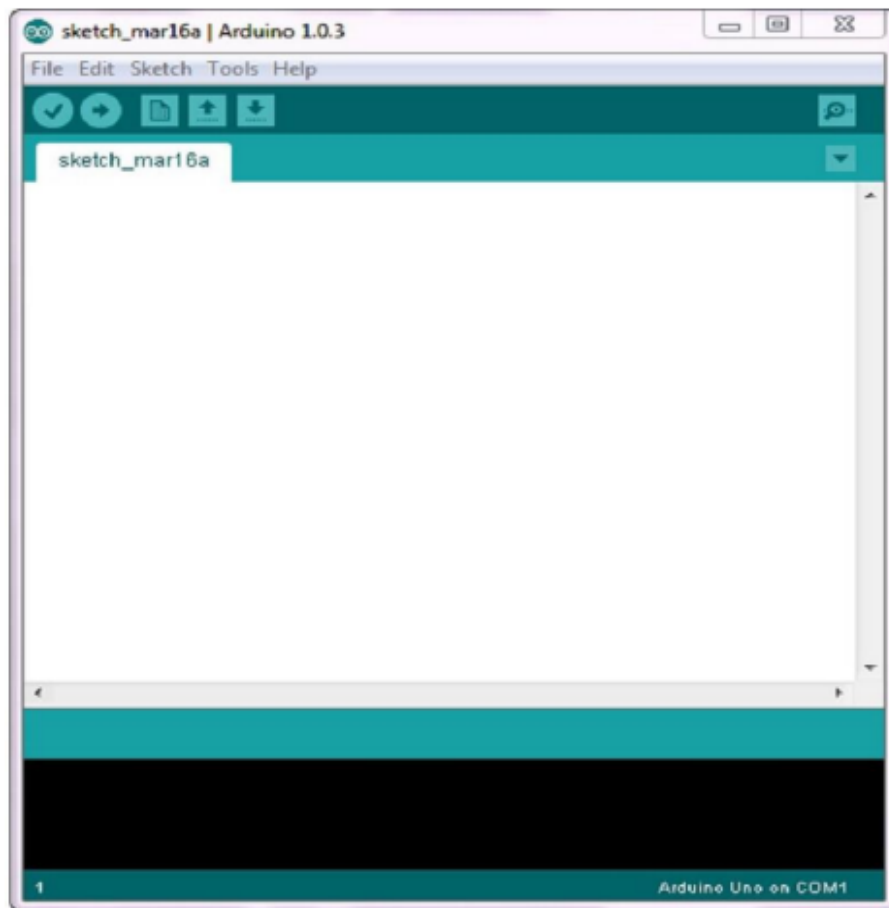
- Bước 7: Các linh kiện đã chọn sẽ xuất hiện trong danh sách thiết bị. Chọn linh kiện và đặt nó vào bảng thiết kế bằng cách nhấp chuột trái. Đặt tất cả các linh kiện cần thiết và định tuyến dây, tức là tạo kết nối. Selection mode ở trên component mode hoặc component mode cho phép kết nối qua dây. Nhấp chuột trái từ chân này sang chân khác để tạo kết nối. Nhấp đúp chuột phải vào dây được kết nối hoặc linh kiện để loại bỏ kết nối hoặc linh kiện tương ứng. Nhấp đúp vào linh kiện để chỉnh sửa thuộc tính của các linh kiện và nhấp vào Ok.
- Bước 8: Sau khi kết nối mạch, bấm vào nút play để chạy mô phỏng. Mô phỏng có thể được thực hiện, tạm dừng hoặc dừng bất kỳ lúc nào

### **2.2.2. Phần mềm Arduino IDE**

Arduino IDE là một phần mềm lập trình được sử dụng để lập trình các bo mạch phát triển Arduino. Nó cho phép người dùng viết và tải xuống mã nguồn từ máy tính của họ vào bo mạch Arduino.

Phần mềm này có giao diện đơn giản và dễ sử dụng, cho phép người dùng lập trình các dự án sử dụng các loại bo mạch Arduino khác nhau, từ những dự án đơn giản như đèn LED cho đến những dự án phức tạp hơn như các robot di động hay hệ thống tự động hóa.

Arduino IDE được phát triển trên nền tảng Java và có thể được cài đặt trên các hệ điều hành như Windows, MacOS và Linux. Nó cũng có tính năng mã nguồn mở, cho phép người dùng tùy chỉnh và phát triển các tính năng mới cho phần mềm.



**Hình 2. 10:** *Arduino IDE*

Arduino Toolbar: có một số button và chức năng của chúng như sau :



Verify : kiểm tra code có lỗi hay không



Upload: nạp code đang soạn thảo vào Arduino



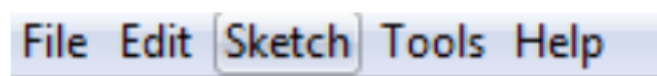
New, Open, Save : Tạo mới, mở và Save sketch



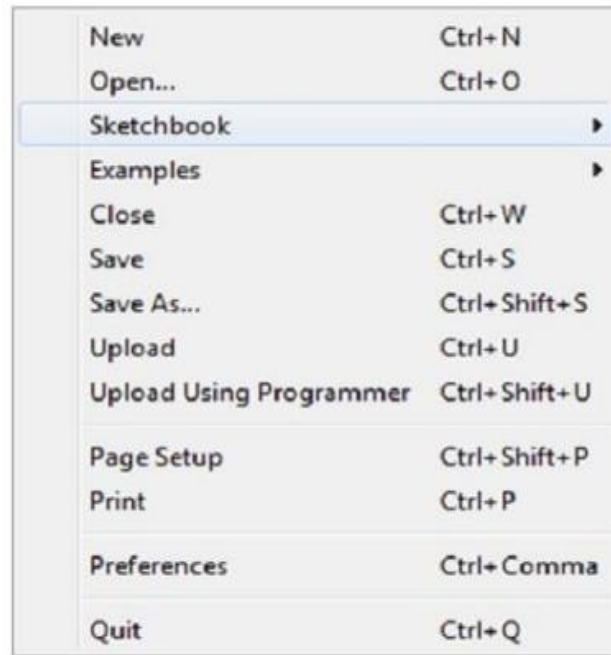
Serial Monitor : Đây là màn hình hiển thị dữ liệu từ Arduino gửi lên máy tính



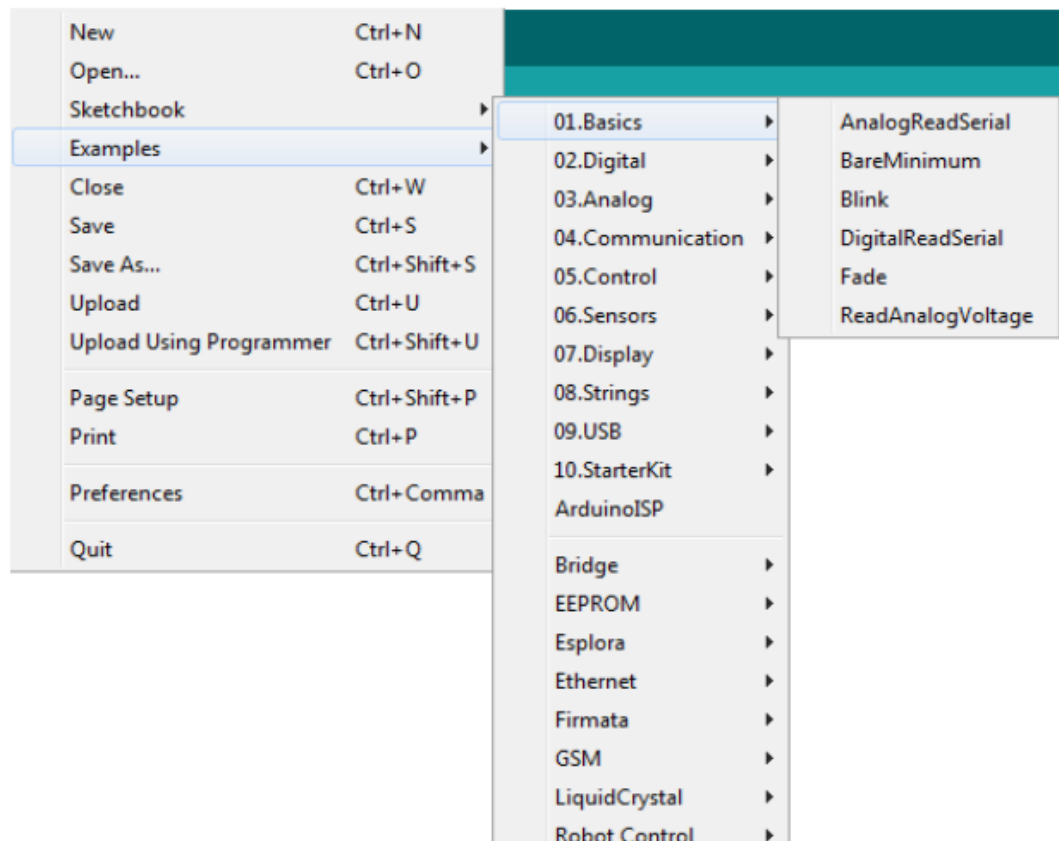
b) Arduino IDE Menu:







Trong file menu chúng ta quan tâm tới mục Examples đây là nơi chứa code mẫu ví dụ như: cách sử dụng các chân digital, analog, sensor...



Edit menu:

Undo	Ctrl+Z
Redo	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Copy for Forum	Ctrl+Shift+C
Copy as HTML	Ctrl+Alt+C
Paste	Ctrl+V
Select All	Ctrl+A
Comment/Uncomment	Ctrl+Slash
Increase Indent	Ctrl+Close Bracket
Decrease Indent	Ctrl+Open Bracket
Find...	Ctrl+F
Find Next	Ctrl+G
Find Previous	Ctrl+Shift+G
Use Selection For Find	Ctrl+E

Sketch menu:

Verify / Compile	Ctrl+R
Show Sketch Folder	Ctrl+K
Add File...	
Import Library...	▶

Trong Sketch menu :

Verify/ Compile : chức năng kiểm tra lỗi code.

Show Sketch Folder : hiển thị nơi code được lưu.

Add File : thêm vào một Tập code mới.

Import Library : thêm thư viện cho IDE.

Tool menu:

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Serial Monitor	Ctrl+Shift+M
Board	▶
Serial Port	▶
Programmer	▶
Burn Bootloader	

Trong Tool menu ta quan tâm các mục Board và Serial Port.

Mục Board : các bạn cần phải lựa chọn bo mạch cho phù hợp với loại bo mà bạn sử dụng nếu là Arduino Uno thì phải chọn như hình:



Nếu các bạn sử dụng loại bo khác thì phải chọn đúng loại bo mà mình đang có nếu sai thì code Upload vào chip sẽ bị lỗi.

Serial Port: đây là nơi lựa chọn cổng Com của Arduino. Khi chúng ta cài đặt driver thì máy tính sẽ hiện thông báo tên cổng Com của Arduino là bao nhiêu, ta chỉ việc vào Serial Port chọn đúng cổng Com để nạp code, nếu chọn sai thì không thể nạp code cho Arduino được.

### **2.2.3. Android Studio**

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày.

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác.

Android Studio khiến việc tạo ứng dụng trở nên dễ dàng hơn đáng kể so với phần mềm không chuyên dụng. Đối với người mới bắt đầu, có rất nhiều thứ để học và nhiều thông tin có sẵn, thậm chí thông qua các kênh chính thức nhưng chúng có thể đã lỗi thời hoặc quá nhiều thông tin khiến họ cảm thấy choáng ngợp. Bài viết này sẽ giải thích ngắn gọn nhưng chi tiết về một số chức năng cơ bản của nó để bạn có thể nắm bắt được bước đầu trong công cuộc phát triển Android của mình.

Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.



**Hình 2. 11:** *Android studio*

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Đặc tả yêu cầu hệ thống

#### 3.1.1. Các yêu cầu chức năng

- Điều khiển từ xa: Hệ thống điều khiển từ xa đáp ứng với tần số sóng radio của bộ điều khiển để đảm bảo việc điều khiển ô tô từ xa một cách chính xác và tin cậy.
- Kiểm soát tốc độ: Ô tô có khả năng điều chỉnh tốc độ di chuyển dựa trên các lệnh điều khiển từ xa để đảm bảo an toàn khi di chuyển.
- Quay đầu: Ô tô có khả năng quay đầu ở chỗ hoặc quay trái, quay phải để thay đổi hướng di chuyển.
- Lùi xe: Ô tô có khả năng di chuyển lùi để thoát khỏi các tình huống khó khăn hoặc những chỗ hẹp.
- Điều khiển hướng lái: Hệ thống điều khiển hướng lái cho phép người điều khiển có khả năng thay đổi hướng di chuyển của ô tô một cách dễ dàng.
- Cảm biến va chạm: Các cảm biến va chạm được tích hợp để ngăn chặn ô tô va chạm với các vật thể hoặc người trong quá trình di chuyển.
- Hệ thống đèn: Hệ thống đèn đủ sáng để người điều khiển có thể quan sát được ô tô trong điều kiện ánh sáng yếu hoặc ban đêm.
- Công suất và thời lượng pin: Ô tô có đủ công suất để hoạt động trong thời gian dài và sử dụng pin có thể thay thế.
- Thiết kế: Thiết kế ô tô phải đáp ứng được các yêu cầu của người dùng, bao gồm thiết kế cấu trúc, kích thước và kiểu dáng của ô tô.

#### 3.1.2. Các yêu cầu phi chức năng

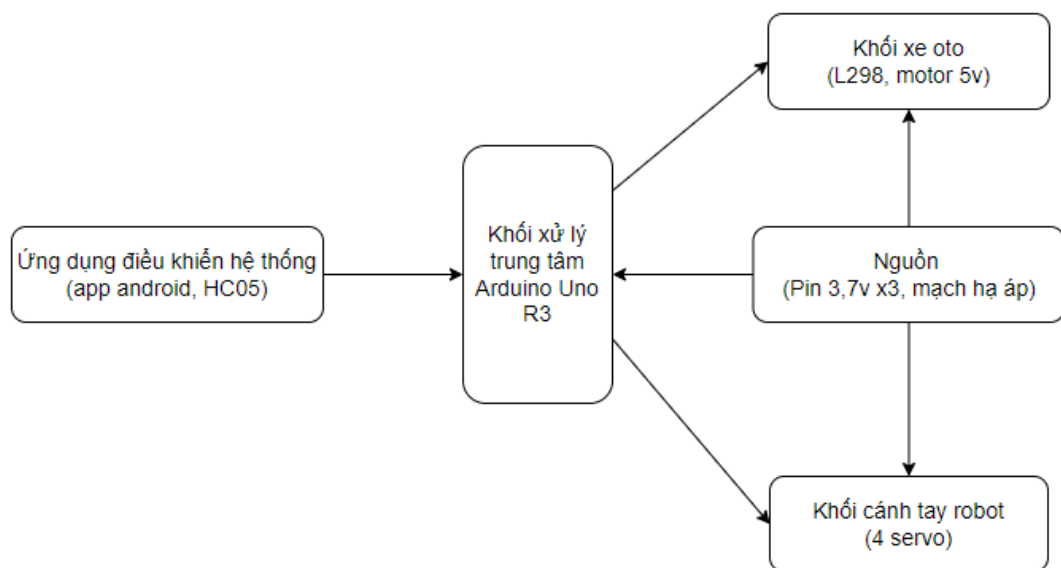
- Tốc độ: Tốc độ của ô tô nên được điều chỉnh từ xa để tránh tai nạn.
- Hướng đi: Ô tô nên có thể được điều khiển để quay và di chuyển theo các hướng khác nhau.

- Tự động dừng: Ô tô nên có khả năng dừng lại khi gặp chướng ngại vật hoặc khi mất tín hiệu điều khiển.
- Tránh va chạm: Ô tô nên có cảm biến để tránh va chạm với các vật cản hoặc tường.
- Điều khiển từ xa: Điều khiển ô tô từ xa bằng một bộ điều khiển được tích hợp trên điện thoại hoặc máy tính.
- Cảm biến: Ô tô nên được trang bị các cảm biến để phát hiện và tránh các vật cản.
- Pin: Ô tô nên sử dụng một loại pin dễ thay thế và có thời lượng sử dụng lâu dài.
- Khả năng điều chỉnh: Ô tô nên có khả năng điều chỉnh để tùy chỉnh tốc độ, độ xoay và độ nhạy của cảm biến.
- Tiết kiệm năng lượng: Ô tô nên được thiết kế để tiết kiệm năng lượng và giảm thiểu khả năng hao phí điện năng.

### 3.2. Thiết kế hệ thống

#### 3.2.1. Thiết kế phần cứng cho hệ thống

##### ❖ Sơ đồ khối



**Hình 3. 1:** Sơ đồ khối

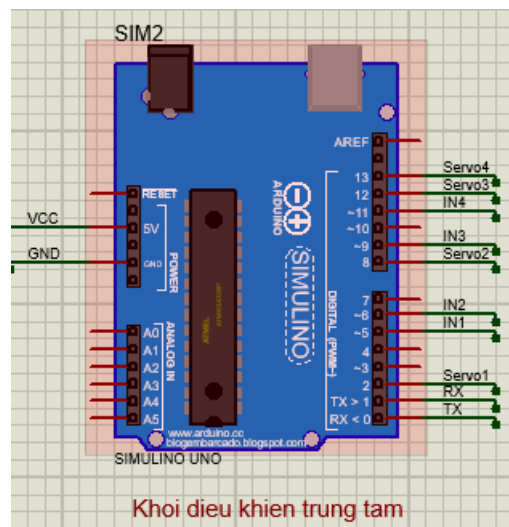


Sơ đồ nguyên lý bao gồm các linh kiện điện tử được sử dụng và được phân chia thành các khối chức năng như: Khối điều khiển trung tâm (arduino), khối điều khiển motor (4 bánh xe), khối cánh tay robot (4 servor), khối điều khiển bluetooth.

Tất cả các khối được kết nối với nhau chặt chẽ để giải quyết yêu cầu của đề tài.

❖ **Phân tích mạch sơ đồ nguyên lý:**

- Khối xử lý trung tâm (arduino): Nhận lệnh điều khiển từ modul HC05 gửi tới từ đó, đưa ra các lệnh điều khiển các thiết bị L298 điều khiển xe (tiền, lùi, quay trái, quay phải, tăng tốc, dừng,...) và điều khiển cánh tay robot gắp các vật thể.



**Hình 3. 3:** Khối điều khiển trung tâm

- Khối điều khiển động cơ motor (4 bánh xe): nhận lệnh điều khiển từ khối điều khiển trung tâm (arduino uno).

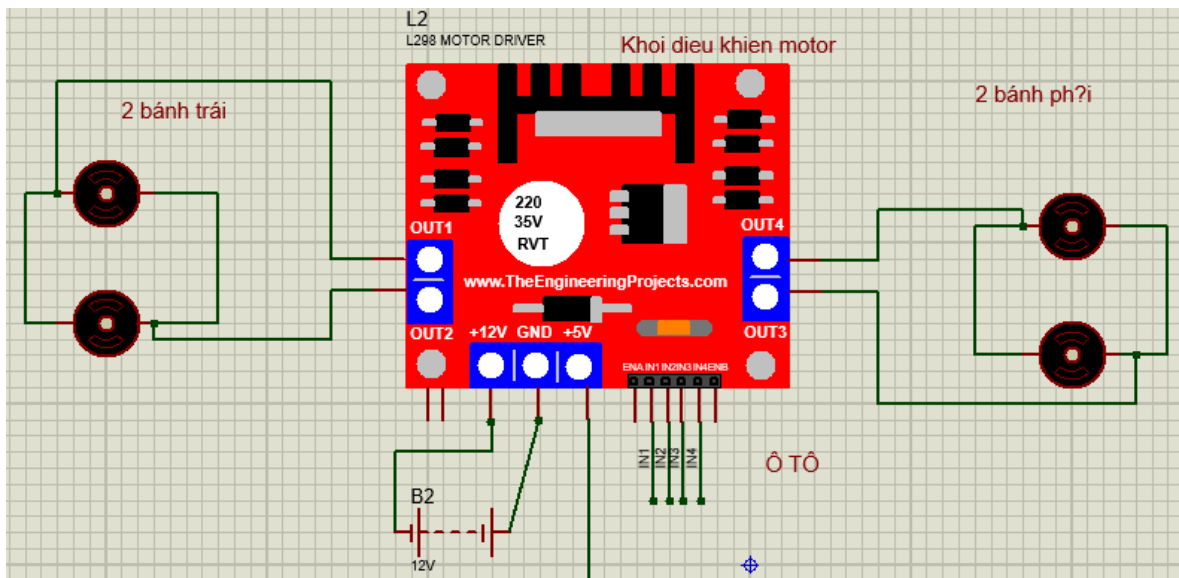
+ Tiến: arduino gửi tín hiệu đi thẳng tới L298. Lệnh điều khiển cho 4 bánh xe có cùng tốc độ và quay cùng chiều là chiều kim đồng hồ, tức là 2 bánh trái nhận lệnh từ chân uot1, uot2 và 2 bánh phải nhận lệnh từ chân uot3, uot4 có cùng tín hiệu điều khiển.

+ Lùi: Tiến: arduino gửi tín hiệu đi lùi tới L298. Lệnh điều khiển cho 4 bánh xe có cùng tốc độ và quay cùng chiều là chiều kim đồng hồ, tức là 2 bánh trái nhận lệnh từ chân uot1, uot2 và 2 bánh phải nhận lệnh từ chân uot3, uot4 có cùng tín hiệu điều khiển.



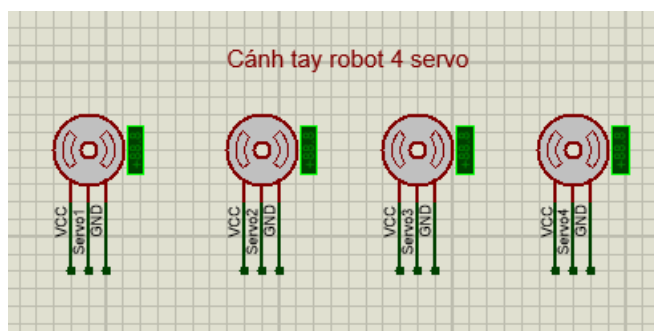
+ quay trái: arduino gửi tín hiệu quay trái tới L298. Lệnh điều khiển cho 4 bánh xe có cùng tốc độ và quay cùng chiều là chiều kim đồng hồ, tức là 2 bánh trái nhận lệnh từ chân uot1, uot2 sẽ đi lùi và 2 bánh phải nhận lệnh từ chân uot3, uot4 sẽ đi tiến.

+ quay phải: arduino gửi tín hiệu quay phải tới L298. Lệnh điều khiển cho 4 bánh xe có cùng tốc độ và quay cùng chiều là chiều kim đồng hồ, tức là 2 bánh trái nhận lệnh từ chân uot1, uot2 sẽ đi tiến và 2 bánh phải nhận lệnh từ chân uot3, uot4 sẽ đi lùi.



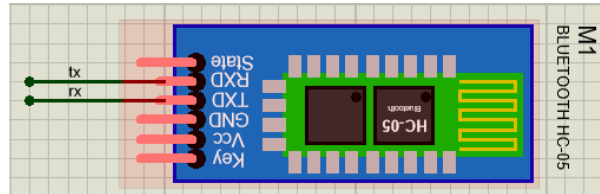
**Hình 3. 4:** Khởi điều khiển động cơ motor

- Khởi điều khiển cánh tay robot: nhận lệnh điều khiển từ khởi điều khiển trung tâm (arduino uno), điều khiển các sevor có góc quay tương ứng với vị trí cần gấp vật thể.



**Hình 3. 5:** Khởi điều khiển cánh tay robot

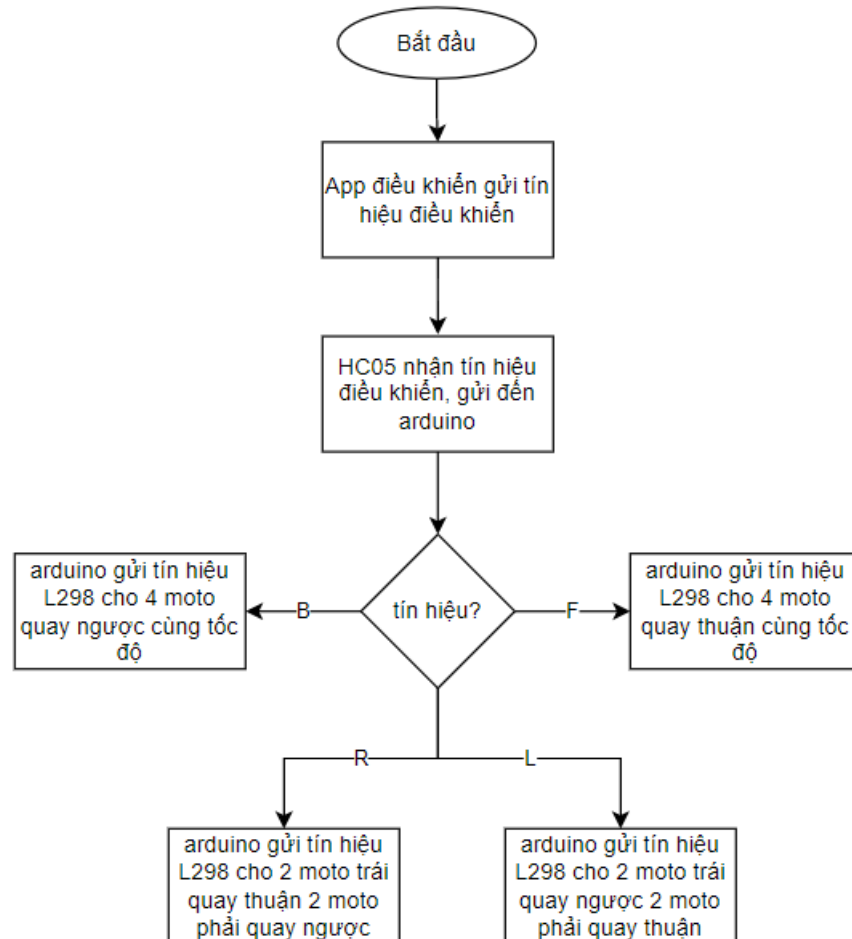
- Khối điều khiển bluetooth: nhận lệnh điều khiển từ app mobile trên điện thoại thông minh được kết bluetooth với modul HC05 và gửi lệnh điều khiển tới khối điều khiển trung tâm để thực hiện tác vụ.



**Hình 3. 6:** Khối điều khiển bluetooth

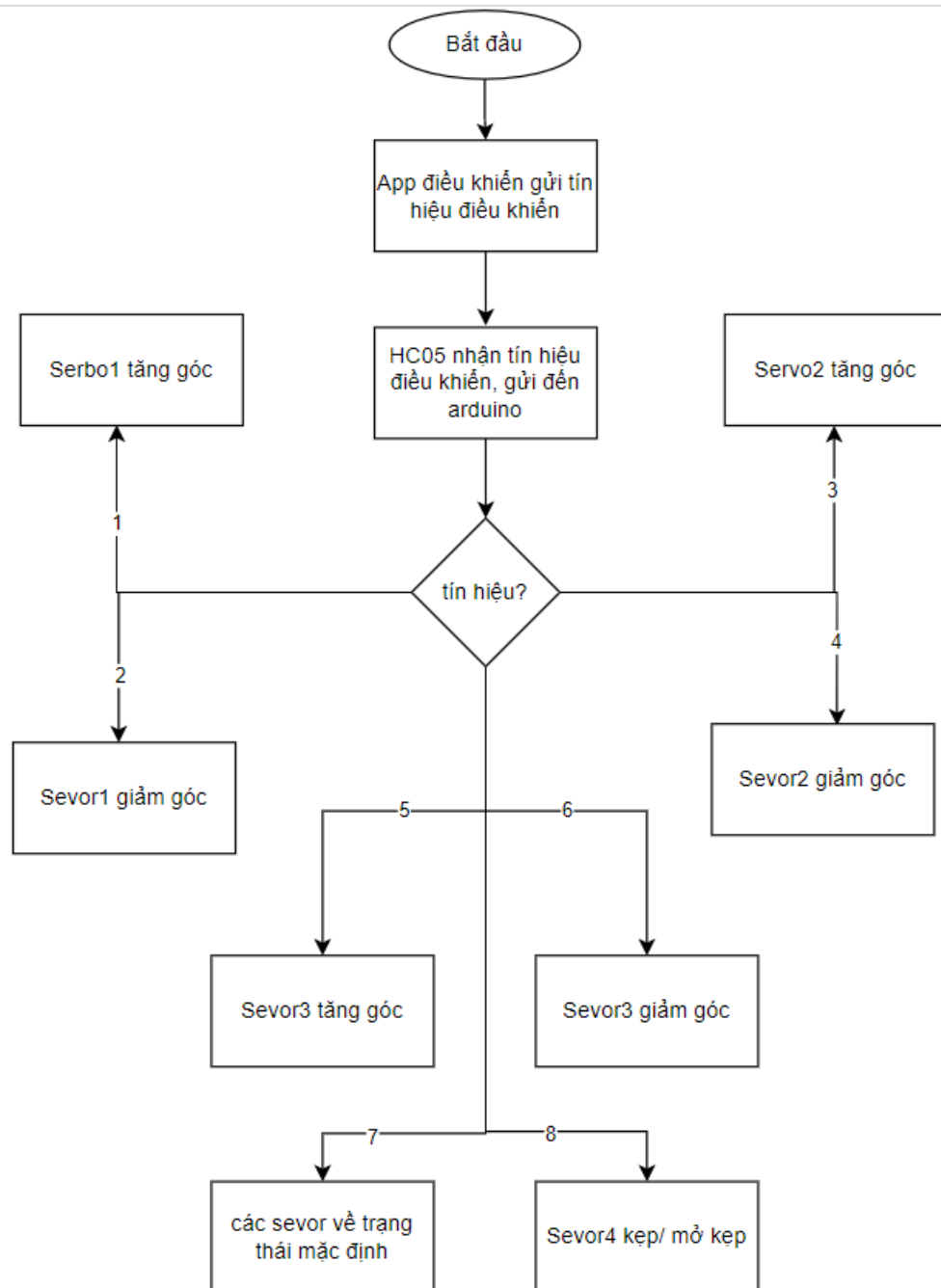
### 3.2.2. Thiết kế phần mềm cho hệ thống

❖ Chức năng điều khiển xe ô tô



**Hình 3. 7:** Điều khiển xe

❖ Điều khiển cánh tay robot



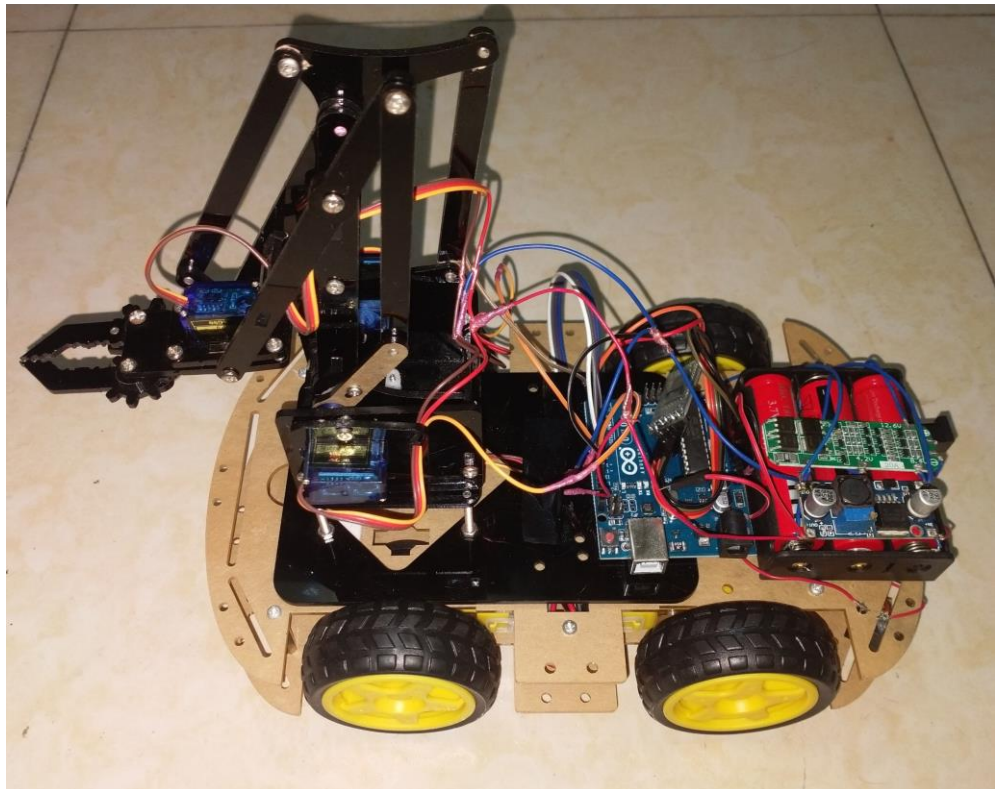
**Hình 3. 8:** Điều khiển cánh tay robot

## CHƯƠNG 4: TÍCH HỢP VÀ ĐÁNH GIÁ HỆ THỐNG

### 4.1. Xây dựng và tích hợp hệ thống

#### 4.1.1. Xây dựng hệ thống

##### a) Xây dựng phần cứng hệ thống



**Hình 3. 9:** Phần cứng hệ thống

- Hệ thống nguồn: x3 pin 3,7 + mạch hạ áp + mạch sạc pin
- Trung tâm điều khiển arduino uno R3
- L298 điều khiển 4 moto
- Bộ khung xe ô tô 4 bánh
- Cánh tay robot: 4 servor
- Modul buetooth HC05

***b) Phần mềm hệ thống***

❖ Code xử lý cho arduino

```
#include <Arduino.h>

#include <Servo.h>

Servo servo_left;

Servo servo_right;

Servo servo_center;

Servo servo_kep;

bool toggle = 1;

byte goc_left; // đọc giá trị hiện tại của servo_left

byte goc_right; // đọc giá trị hiện tại của servo_right

byte goc_center; // đọc giá trị hiện tại của servo_center

#define servoleft 12

#define servoright 13

#define servocenter 8

#define servokep 2

int IN1 = 5; //IN1 L298N

int IN2 = 6; //IN2 L298N

int IN3 = 9; //IN3 L298N

int IN4 = 11; //IN4 L298N

#define Toc_do HIGH

#define aa0 LOW

//int Toc_do = 200; // Toc do dong co (0-255)
```

```
void auto_home(){
    servo_left.write(70); //70 - 150
    servo_right.write(90); //30-180
    servo_center.write(80); //
    servo_kep.write(130); // 30-130
}

void gohome(){ if(flag == '7'){auto_home(); } }

void left(){
    goc_left = servo_left.read();
    if(flag == '1') //Nut A{
        if(goc_left < 150){
            goc_left = goc_left + 1;    servo_left.write(goc_left);    delay(10);
        }
        else servo_left.write(150);
    }
    if(flag == '2') //Nut C{
        if(goc_left > 70){
            goc_left = goc_left - 1;    servo_left.write(goc_left);    delay(10);
        }
        else servo_left.write(70);
    }
    // Serial.print(" - left: "); Serial.print(servo_left.read());
}
```

```
void right(){
    goc_right = servo_right.read();
    if(flag == '3') {
        if(goc_right < 180) {
            goc_right = goc_right + 1;    servo_right.write(goc_right);    delay(10) }
        else servo_right.write(180);
    }
    if(flag == '4') { if(goc_right > 30){
        goc_right = goc_right - 1;    servo_right.write(goc_right);    delay(10); }
        else servo_right.write(30); }
    }
void center(){
    goc_center = servo_center.read();
    if(flag == '5') { if(goc_center < 180) {
        goc_center = goc_center + 1; servo_center.write(goc_center); delay(10); }
        else servo_center.write(180);}
    if(flag == '6') {if(goc_center > 30){
        goc_center = goc_center - 1;    servo_center.write(goc_center);
        delay(10);
        }
        else servo_center.write(30);
    }
}
```

```
int flagg; char flag = 'S';

void loop() {

  if (Serial.available()){   flagg = Serial.read(); flag = (char)flagg;  }

  if (flag == 'L') { //re trai

    digitalWrite(IN4, Toc_do);   digitalWrite(IN2, aa0);

    digitalWrite(IN3, aa0); digitalWrite(IN1, Toc_do);

  }

  if (flag == 'R') { //re phai

    digitalWrite(IN4, aa0); digitalWrite(IN2, Toc_do);

    digitalWrite(IN3, Toc_do);digitalWrite(IN1, aa0);

  }

  if (flag == 'S') { //Dung

    digitalWrite(IN4, aa0);digitalWrite(IN2, aa0);

    digitalWrite(IN3, aa0);digitalWrite(IN1, aa0); }

  if (flag == 'F') { //chay thuan

    digitalWrite(IN4, Toc_do);digitalWrite(IN2, Toc_do);

    digitalWrite(IN3, aa0);digitalWrite(IN1, aa0);  }

  if (flag == 'B') { //Chay nguoc

    digitalWrite(IN4, aa0); digitalWrite(IN2, aa0);

    digitalWrite(IN3, Toc_do); digitalWrite(IN1, Toc_do);

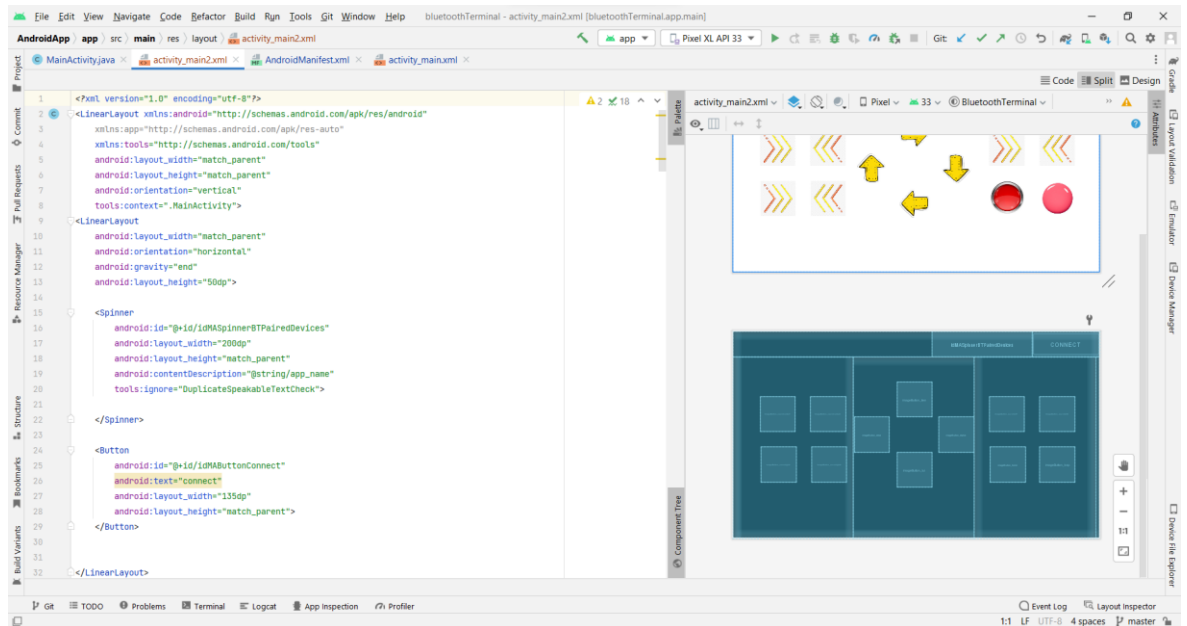
  }

  left(); right(); center(); control_kep(); gohome();

}
```



### ❖ Lập trình app điều khiển



Hình 3. 10: Lập trình app android điều khiển

- Lập trình giao diện app, file .xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:gravity="end"
        android:layout_height="50dp">
        <Spinner
            android:id="@+id/idMASpinnerBTPairedDevices"
            android:layout_width="200dp"
            android:layout_height="match_parent"
            android:contentDescription="@string/app_name"
            tools:ignore="DuplicateSpeakableTextCheck">
        </Spinner>
        <Button
            android:id="@+id/idMAButtonConnect"
            android:text="connect"
            android:layout_width="135dp"
            android:layout_height="match_parent">
        </Button>
```

```
</LinearLayout>
<LinearLayout
    android:weightSum="10"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false">
    <LinearLayout
        android:layout_width="0dp"
        android:layout_weight="3.3"
        android:layout_height="match_parent">
        <ImageButton
            android:id="@+id/imageButton_sevorcenter1"
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_marginStart="55dp"
            android:layout_marginTop="80dp"
            android:background="@color/white"
            android:contentDescription="@string/app_name"
            android:rotation="0"
            android:scaleType="fitXY"
            android:src="@drawable/muitenserver2"
            tools:ignore="ImageContrastCheck">
        </ImageButton>
        <ImageButton
            android:id="@+id/imageButton_sevorcenter2"
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_marginStart="30dp"
            android:layout_marginTop="80dp"
            android:background="@color/white"
            android:contentDescription="@string/app_name"
            android:rotation="180"
            android:scaleType="fitXY"
            android:src="@drawable/muitenserver2"
            tools:ignore="ImageContrastCheck">
        </ImageButton>
        <ImageButton
            android:id="@+id/imageButton_sevorright1"
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_marginStart="-170dp"
            android:layout_marginTop="180dp"
            android:background="@color/white"
            android:contentDescription="@string/app_name"
            android:rotation="0"
            android:scaleType="fitXY"
            android:src="@drawable/muitenserver2"
            tools:ignore="ImageContrastCheck">
        </ImageButton>
        <ImageButton
            android:id="@+id/imageButton_sevorright2"
```

```
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginStart="30dp"
        android:layout_marginTop="180dp"
        android:background="@color/white"
        android:contentDescription="@string/app_name"
        android:rotation="180"
        android:scaleType="fitXY"
        android:src="@drawable/muitenserver2"
        tools:ignore="ImageContrastCheck">
    </ImageButton>
</LinearLayout>
<LinearLayout
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:layout_width="0dp"
    android:layout_weight="3.3"
    android:layout_height="match_parent">
    <ImageButton
        android:id="@+id/imageButton_tien"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginStart="2dp"
        android:layout_marginTop="50dp"
        android:background="@color/white"
        android:contentDescription="@string/app_name"
        android:rotation="270"
        android:scaleType="fitXY"
        android:src="@drawable/muitenoto1">
    </ImageButton>
    <ImageButton
        android:id="@+id/imageButton_retrai"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginStart="-85dp"
        android:background="@color/white"
        android:contentDescription="@string/app_name"
        android:rotation="180"
        android:scaleType="fitXY"
        android:src="@drawable/muitenoto1">
    </ImageButton>
    <ImageButton
        android:id="@+id/imageButton_rephai"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginStart="85dp"
        android:layout_marginTop="-70dp"
        android:background="@color/white"
        android:contentDescription="@string/app_name"
        android:rotation="0"
        android:scaleType="fitXY"
```

```
        android:src="@drawable/muitenoto1">
</ImageButton>
<ImageButton
    android:id="@+id/imageButton_lui"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginStart="2dp"
    android:background="@color/white"
    android:contentDescription="@string/app_name"
    android:rotation="90"
    android:scaleType="fitXY"
    android:src="@drawable/muitenoto1">
</ImageButton>
</LinearLayout>
<LinearLayout
    android:layout_width="0dp"
    android:layout_weight="3.3"
    android:layout_height="match_parent">
<ImageButton
    android:id="@+id/imageButton_sevorleft1"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginStart="30dp"
    android:layout_marginTop="80dp"
    android:background="@color/white"
    android:contentDescription="@string/app_name"
    android:rotation="0"
    android:scaleType="fitXY"
    android:src="@drawable/muitenservor2"
    tools:ignore="ImageContrastCheck">
</ImageButton>
<ImageButton
    android:id="@+id/imageButton_sevorleft2"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginStart="30dp"
    android:layout_marginTop="80dp"
    android:background="@color/white"
    android:contentDescription="@string/app_name"
    android:rotation="180"
    android:scaleType="fitXY"
    android:src="@drawable/muitenservor2"
    tools:ignore="ImageContrastCheck">
</ImageButton>
<ImageButton
    android:id="@+id/imageButton_home"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginStart="-170dp"
    android:layout_marginTop="180dp"
    android:background="@color/white"
```

```
        android:contentDescription="@string/app_name"
        android:rotation="180"
        android:scaleType="fitXY"
        android:src="@drawable/home">
    </ImageButton>
    <ImageButton
        android:id="@+id/imageButton_kep"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginStart="30dp"
        android:layout_marginTop="180dp"
        android:background="@color/white"
        android:contentDescription="@string/app_name"
        android:rotation="0"
        android:scaleType="fitXY"
        android:src="@drawable/kep"
        tools:ignore="ImageContrastCheck">
    </ImageButton>
</LinearLayout>
</LinearLayout>
</LinearLayout>
```

### - Lập trình flie .java

```
package com.exloringembeddedsystems.bluetoothterminal;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.Spinner;
import android.widget.Toast;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.Set;
import java.util.UUID;
```

```
public class MainActivity extends AppCompatActivity {
    private static final String TAG = "DEBUG_MA" ;
    Button buttonBTConnect;
    ImageButton button_Tien;
    ImageButton button_Lui;
    ImageButton button_ReTrai;
    ImageButton button_RePhai;
    ImageButton button_sevorCenter_1;
    ImageButton button_sevorCenter_2;
    ImageButton button_sevorRight_1;
    ImageButton button_sevorRight_2;
    ImageButton button_sevorLeft_1;
    ImageButton button_sevorLeft_2;
    ImageButton button_sevorKep;
    ImageButton button_home;

    Spinner spinnerBTPairedDevices;

    static final UUID MY_UUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");
    BluetoothSocket BSocket = null;
    BluetoothAdapter BTAdaptor = null;
    Set<BluetoothDevice> BTPairedDevices = null;
    boolean bBTConnected = false;
    BluetoothDevice BTDevice = null;
    classBTInitDataCommunication cBTInitSendReceive =null;

    static public final int BT_CON_STATUS_NOT_CONNECTED    =0;
    static public final int BT_CON_STATUS_CONNECTING      =1;
    static public final int BT_CON_STATUS_CONNECTED       =2;
    static public final int BT_CON_STATUS_FAILED          =3;
    static public final int BT_CON_STATUS_CONNECTION_LOST =4;
    static public int iBTConnectionStatus =
BT_CON_STATUS_NOT_CONNECTED;

    static final int BT_STATE_LISTENING                    =1;
    static final int BT_STATE_CONNECTING                   =2;
    static final int BT_STATE_CONNECTED                    =3;
    static final int BT_STATE_CONNECTION_FAILED            =4;
    static final int BT_STATE_MESSAGE_RECEIVED             =5;

    String data = null;
    String datastop = "S";

    @SuppressWarnings("ClickableViewAccessibility")
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    Log.d(TAG, "onCreate-Start");

    spinnerBTPairedDevices =
    findViewById(R.id.idMASpinnerBTPairedDevices);
    buttonBTConnect = findViewById(R.id.idMAButtonConnect);

    button_Tien = findViewById(R.id.imageButton_tien);
    button_Lui = findViewById(R.id.imageButton_lui);
    button_ReTrai = findViewById(R.id.imageButton_retrai);
    button_RePhai = findViewById(R.id.imageButton_rephai);
    button_sevorCenter_1 =
    findViewById(R.id.imageButton_sevorcenter1);
    button_sevorCenter_2 =
    findViewById(R.id.imageButton_sevorcenter2);
    button_sevorRight_1 =
    findViewById(R.id.imageButton_sevorright1);
    button_sevorRight_2 =
    findViewById(R.id.imageButton_sevorright2);
    button_sevorLeft_1 =
    findViewById(R.id.imageButton_sevorleft1);
    button_sevorLeft_2 =
    findViewById(R.id.imageButton_sevorleft2);
    button_sevorKep = findViewById(R.id.imageButton_kep);
    button_home = findViewById(R.id.imageButton_home);

    button_Tien.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            if (event.getAction() == MotionEvent.ACTION_DOWN) {
                data = "F";
                sendMessage(data);
                Log.d(TAG, "Button Click Tien");
                Log.d(TAG, " " + button_Tien.getWidth());
                button_Tien.setScaleX(0.9f);
                button_Tien.setScaleY(0.9f);
                return true;
            } else if (event.getAction() ==
    MotionEvent.ACTION_UP) {
                Log.d(TAG, "Button Click Tien _____");
                sendMessage(datastop);
                button_Tien.setScaleX(1f);
                button_Tien.setScaleY(1f);
                return true;
            }
        }
    })
}
```

```
        return false;
    }
});

button_Lui.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "B";
            Log.d(TAG, "Button Click Lui");
            sendMessage(data);
            button_Lui.setScaleX(0.9f);
            button_Lui.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            sendMessage(datastop);
            Log.d(TAG, "Button Click Lui_____");
            button_Lui.setScaleX(1f);
            button_Lui.setScaleY(1f);
            return true;
        }
        return false;
    }
});

button_RePhai.setOnTouchListener(new View.OnTouchListener()
{
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "R";
            Log.d(TAG, "Button Click Phai");
            sendMessage(data);
            button_RePhai.setScaleX(0.9f);
            button_RePhai.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            sendMessage(datastop);
            button_RePhai.setScaleX(1f);
            button_RePhai.setScaleY(1f);
            Log.d(TAG, "Button Click Phai_____");
            return true;
        }
        return false;
    }
});

button_ReTrai.setOnTouchListener(new View.OnTouchListener()
```



```
{  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            data = "L";  
            sendMessage(data);  
            Log.d(TAG, "Button Click Trai");  
            button_ReTrai.setScaleX(0.9f);  
            button_ReTrai.setScaleY(0.9f);  
            return true;  
        } else if (event.getAction() ==  
MotionEvent.ACTION_UP) {  
            sendMessage(datastop);  
            Log.d(TAG, "Button Click Trai_____");  
            button_ReTrai.setScaleX(1f);  
            button_ReTrai.setScaleY(1f);  
            return true;  
        }  
        return false;  
    }  
};  
  
    button_sevorCenter_1.setOnTouchListener(new  
View.OnTouchListener() {  
        @Override  
        public boolean onTouch(View v, MotionEvent event) {  
            if (event.getAction() == MotionEvent.ACTION_DOWN) {  
                data = "5";  
                sendMessage(data);  
                Log.d(TAG, "Button Click servo center1");  
                button_sevorCenter_1.setScaleX(0.9f);  
                button_sevorCenter_1.setScaleY(0.9f);  
                return true;  
            } else if (event.getAction() ==  
MotionEvent.ACTION_UP) {  
                Log.d(TAG, "Button Click servo center2");  
                sendMessage(datastop);  
                button_sevorCenter_1.setScaleX(1f);  
                button_sevorCenter_1.setScaleY(1f);  
                return true;  
            }  
            return false;  
        }  
    });  
  
    button_sevorCenter_2.setOnTouchListener(new  
View.OnTouchListener() {  
        @Override  
        public boolean onTouch(View v, MotionEvent event) {  
            if (event.getAction() == MotionEvent.ACTION_DOWN) {  
                data = "6";
```

```
        sendMessage(data);
        Log.d(TAG, "Button Click Tien");
        button_sevorCenter_2.setScaleX(0.9f);
        button_sevorCenter_2.setScaleY(0.9f);
        return true;
    } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
        Log.d(TAG, "Button Click Tien _____");
        sendMessage(datastop);
        button_sevorCenter_2.setScaleX(1f);
        button_sevorCenter_2.setScaleY(1f);
        return true;
    }
    return false;
}
});

button_sevorRight_1.setOnTouchListener(new
View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "3";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_sevorRight_1.setScaleX(0.9f);
            button_sevorRight_1.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_sevorRight_1.setScaleX(1f);
            button_sevorRight_1.setScaleY(1f);
            return true;
        }
        return false;
    }
});

button_sevorRight_2.setOnTouchListener(new
View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "4";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_sevorRight_2.setScaleX(0.9f);
            button_sevorRight_2.setScaleY(0.9f);
            return true;
        }
    }
});
```

```
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_sevorRight_2.setScaleX(1f);
            button_sevorRight_2.setScaleY(1f);
            return true;
        }
        return false;
    }
});

button_sevorLeft_1.setOnTouchListener(new
View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "1";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_sevorLeft_1.setScaleX(0.9f);
            button_sevorLeft_1.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_sevorLeft_1.setScaleX(1f);
            button_sevorLeft_1.setScaleY(1f);
            return true;
        }
        return false;
    }
});

button_sevorLeft_2.setOnTouchListener(new
View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "2";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_sevorLeft_2.setScaleX(0.9f);
            button_sevorLeft_2.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_sevorLeft_2.setScaleX(1f);
```

```
        button_sevorLeft_2.setScaleY(1f);
        return true;
    }
    return false;
}
});

button_home.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "7";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_home.setScaleX(0.9f);
            button_home.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_home.setScaleX(1f);
            button_home.setScaleY(1f);
            return true;
        }
        return false;
    }
});

button_sevorKep.setOnTouchListener(new
View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            data = "8";
            sendMessage(data);
            Log.d(TAG, "Button Click Tien");
            button_sevorKep.setScaleX(0.9f);
            button_sevorKep.setScaleY(0.9f);
            return true;
        } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
            Log.d(TAG, "Button Click Tien _____");
            sendMessage(datastop);
            button_sevorKep.setScaleX(1f);
            button_sevorKep.setScaleY(1f);
            return true;
        }
        return false;
    }
});
});
```

```
        buttonBTConnect.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Log.d(TAG, "Button Click buttonBTConnect");

        if(bBTConnected == false) {
            if
(spinnerBTPairedDevices.getSelectedItemsPosition() == 0) {
                Log.d(TAG, "Please select BT device");
                Toast.makeText(getApplicationContext(),
                "Please select Bluetooth Device", Toast.LENGTH_SHORT).show();
                return;
            }

            String sSelectedDevice =
spinnerBTPairedDevices.getSelectedItems().toString();
            Log.d(TAG, "Selected device = " +
sSelectedDevice);

            for (BluetoothDevice BTDev : BTPairedDevices) {
                if
(sSelectedDevice.equals(BTDev.getName())) {
                    BTDevice = BTDev;
                    Log.d(TAG, "Selected device UUID = " +
BTDevice.getAddress());

                    cBluetoothConnect cBTConnect = new
cBluetoothConnect(BTDevice);
                    cBTConnect.start();
                }
            }
        }
        else {
            Log.d(TAG, "Disconnecting BTConnection");
            if(BTSocket!=null && BTSocket.isConnected())
            {
                try {
                    BTSocket.close();
                } catch (IOException e) {
                    e.printStackTrace();
                    Log.d(TAG, "BTDisconnect Exp " +
e.getMessage());
                }
            }
            buttonBTConnect.setText("Connect");
        }
    }
}
```

```
        bBTConnected = false;

    }

    }

    });
}

public class cBluetoothConnect extends Thread
{

    public cBluetoothConnect (BluetoothDevice BTDevice)
    {
        Log.i(TAG, "classBTConnect-start");

        try{
            BtSocket =
BTDevice.createRfcommSocketToServiceRecord(MY_UUID);
        }
        catch (Exception exp)
        {
            Log.e(TAG, "classBTConnect-exp" +
exp.getMessage());
        }
    }

    public void run()
    {
        try {
            BtSocket.connect();
            Message message=Message.obtain();
            message.what=BT_STATE_CONNECTED;
            handler.sendMessage(message);

        } catch (IOException e) {
            e.printStackTrace();
            Message message=Message.obtain();
            message.what=BT_STATE_CONNECTION_FAILED;
            handler.sendMessage(message);
        }
    }

}

public class classBTInitDataCommunication extends Thread
```

```
{
    private InputStream inputStream = null;
    private OutputStream outputStream = null;

    public class BTInitDataCommunication (BluetoothSocket
socket)
    {
        Log.i(TAG, "classBTInitDataCommunication-start");

        try {
            inputStream = socket.getInputStream();
            outputStream = socket.getOutputStream();
        } catch (IOException e) {
            e.printStackTrace();
            Log.e(TAG, "classBTInitDataCommunication-start, exp
" + e.getMessage());
        }

    }

    @SuppressWarnings("SetTextI18n")
    public void run()
    {
        byte[] buffer = new byte[1024];
        int bytes;

        while (BTSocket.isConnected())
        {
            try {
                bytes = inputStream.read(buffer);

                handler.obtainMessage(BT_STATE_MESSAGE_RECEIVED, bytes, -
1, buffer).sendToTarget();
            } catch (IOException e) {
                e.printStackTrace();
                Log.e(TAG, "BT disconnect from decide end, exp
" + e.getMessage());

                iBTConnectionStatus = BT_CON_STATUS_CONNECTION_LOST;
                try {
                    //disconnect bluetooth
                    Log.d(TAG, "Disconnecting BTConnection");
                    if (BTSocket != null &&
BTSocket.isConnected())
                    {
                        BTSocket.close();
                    }
                    buttonBTConnect.setText("Connect");
                }
            }
        }
    }
}
```

```
        bBTConnected = false;
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

}

}

public void write(byte[] bytes)
{
    try {
        outputStream.write(bytes);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

Handler handler =new Handler(new Handler.Callback() {
    @Override
    public boolean handleMessage(Message msg) {

        switch (msg.what)
        {
            case BT_STATE_LISTENING:
                Log.d(TAG, "BT_STATE_LISTENING");
                break;
            case BT_STATE_CONNECTING:
                iBTConnectionStatus = BT_CON_STATUS_CONNECTING;
                buttonBTConnect.setText("Connecting..");
                Log.d(TAG, "BT_STATE_CONNECTING");
                break;
            case BT_STATE_CONNECTED:

                iBTConnectionStatus = BT_CON_STATUS_CONNECTED;

                Log.d(TAG, "BT_CON_STATUS_CONNECTED");
                buttonBTConnect.setText("Disconnect");

                cBTInitSendReceive = new
classBTInitDataCommunication(BTSocket);
                cBTInitSendReceive.start();

                bBTConnected = true;
                break;
            case BT_STATE_CONNECTION_FAILED:

                iBTConnectionStatus = BT_CON_STATUS_FAILED;
```



```
Log.d(TAG, "BT_STATE_CONNECTION_FAILED");
bBTConnected = false;
break;

case BT_STATE_MESSAGE_RECEIVED:
    byte[] readBuff= (byte[]) msg.obj;
    String tempMsg=new String(readBuff,0,msg.arg1);
    Log.d(TAG, "Message receive ( " +
tempMsg.length() + " ) data : " + tempMsg);

//                tvReceivedMessage.append(tempMsg);

break;

    }
    return true;
}
});

public void sendMessage(String sMessage)
{
    if( BtSocket!= null &&
iBTConnectionStatus==BT_CON_STATUS_CONNECTED)
    {
        if(BtSocket.isConnected() )
        {
            try {
                cBTInitSendReceive.write(sMessage.getBytes());

//                tvReceivedMessage.append("\r\n-> " +
sMessage);

            }
            catch (Exception exp)
            {

            }
        }
    }
    else {
        Toast.makeText(getApplicationContext(), "Please connect
to
bluetooth", Toast.LENGTH_SHORT).show();

//                tvReceivedMessage.append("\r\n Not connected to
bluetooth");
    }

}
```

```
void getBTPairedDevices()
{
    Log.d(TAG, "getBTPairedDevices - start ");
    BTAdaptor = BluetoothAdapter.getDefaultAdapter();
    if(BTAdaptor == null)

    {
        Log.e(TAG, "getBTPairedDevices , BTAdaptor null ");
        //          editTextSendMessage.setText("\nNo Bluetooth Device in
the phone");
        return;
    }
    else if(!BTAdaptor.isEnabled())
    {
        Log.e(TAG, "getBTPairedDevices , BT not enabled");

        //          editTextSendMessage.setText("\nPlease turn ON
Bluetooth");
        return;
    }

    BTPairedDevices = BTAdaptor.getBondedDevices();

    Log.d(TAG, "getBTPairedDevices , Paired devices count = " +
BTPairedDevices.size());

    for (BluetoothDevice BTDev : BTPairedDevices)
    {
        Log.d(TAG, BTDev.getName() + ", " +
BTDev.getAddress());
    }

}

void populateSpinnerWithBTPairedDevices()
{
    ArrayList<String> alPairedDevices = new ArrayList<>();
    alPairedDevices.add("Select");
    for (BluetoothDevice BTDev : BTPairedDevices)
    {
        alPairedDevices.add(BTDev.getName());
    }
    final ArrayAdapter<String> aaPairedDevices = new
ArrayAdapter<String>(this, R.layout.support_simple_spinner_dropdown_
item, alPairedDevices);

    aaPairedDevices.setDropDownViewResource(R.layout.support_simple_spi
nner_dropdown_item);
}
```

```
        spinnerBTPairedDevices.setAdapter(aaPairedDevices);
    }

    @Override
    protected void onResume() {
        super.onResume();

        Log.d(TAG, "onResume-Resume");

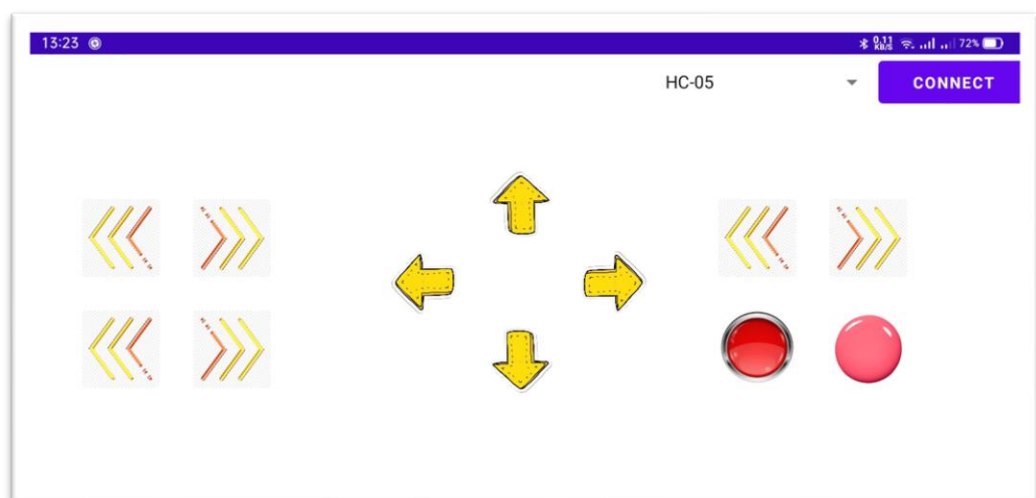
        getBTPairedDevices();
        populateSpinnerWithBTPairedDevices();
        //        readAllData();
    }

    @Override
    protected void onPause() {
        super.onPause();

        Log.d(TAG, "onPause-Start");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        Log.d(TAG, "onDestroy-Start");
    }
}
```

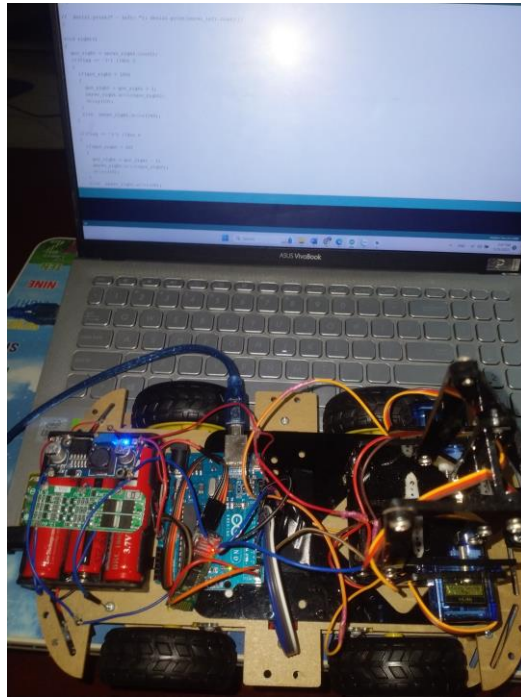


**Hình 3. 11:** Giao diện app điều khiển

#### 4.1.2. Tích hợp giữa phần cứng hệ thống và phần mềm

##### ❖ Nạp code cho arduino

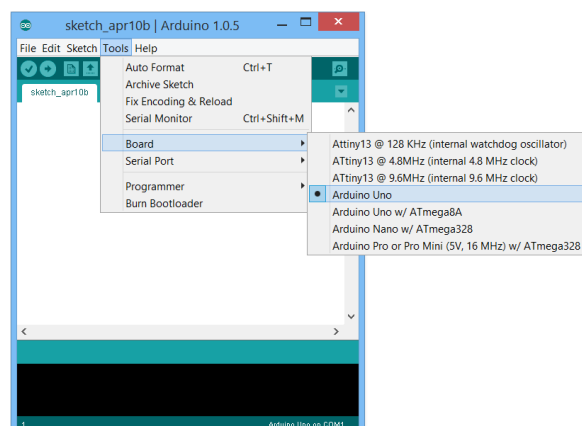
- Kết nối arduino với máy tính



**Hình 3. 12:** Kết nối arduino với máy tính

- Cấu hình phiên bản làm việc:

Vào menu Tools -> Board -> chọn **Arduino Uno**.



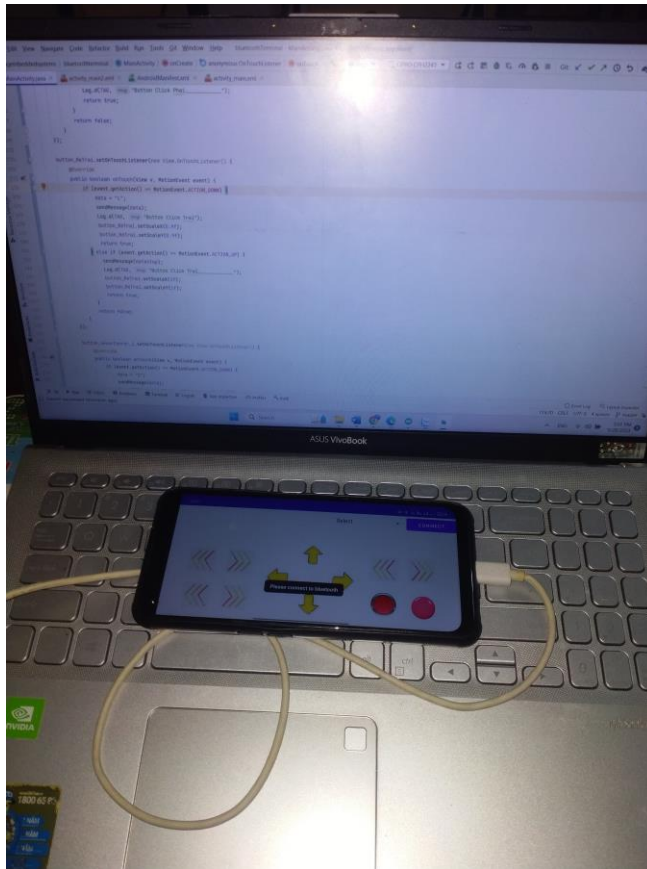
Vào menu Tools -> Serial Port -> chọn cổng Arduino đang kết nối với máy tính.

- Nhấn upload để nạp code xuống arduino uno r3.

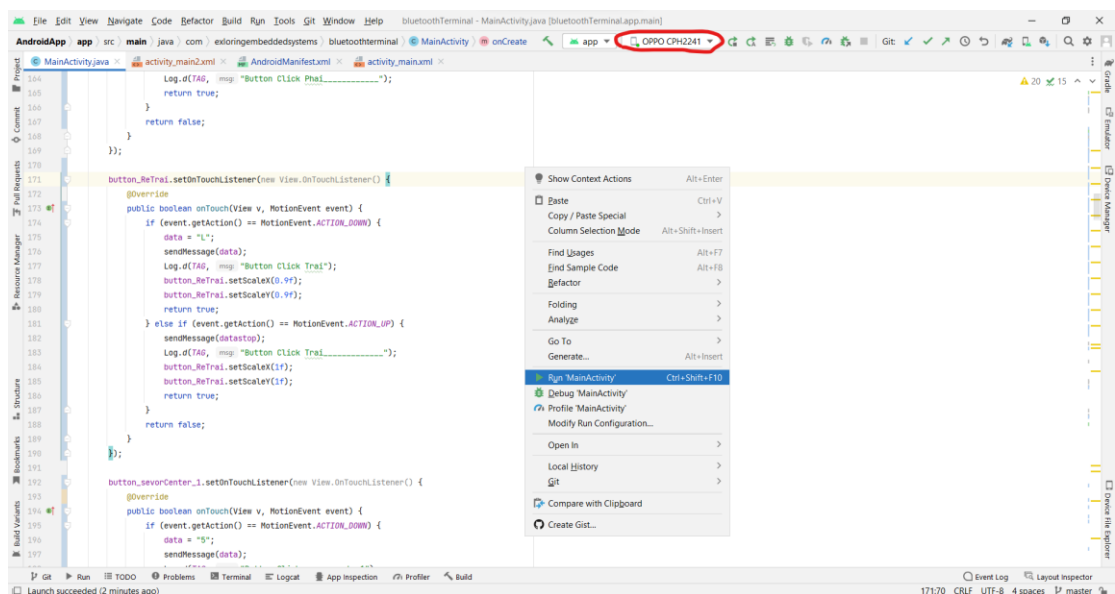
## Thiết kế robo ô tô điều khiển từ xa

### ❖ Tải ứng dụng từ máy tính xuống điện thoại

- Kết nối điện thoại với máy tính.



- Chọn đúng thiết bị kết nối → run app.



## 4.2. Kiểm thử và đánh giá hệ thống

### ❖ Phương pháp kiểm thử và đánh giá hệ thống :

- Sử dụng phần mềm Proteus để mô phỏng và vận hành thử chương trình
- Kiểm tra các linh kiện điện tử có bị lỗi : mất chân , chập cháy , rỉ sét ... hay không.
- Kiểm tra lắp đặt , nối dây phải đúng với sơ đồ mạch ban đầu.
- Vận hành hệ thống , xem chi tiết từng thành phần trong hệ thống có được hoạt động như ý hay không.
- Thiết lập kết nối giữa ứng dụng Android và Arduino thông qua Bluetooth HC05.
- Kiểm tra xem xe robot có nhận được tín hiệu điều khiển từ ứng dụng Android không. Đảm bảo rằng các tín hiệu nhận được được chuyển đổi và xử lý đúng trên Arduino để điều khiển động cơ và servo motor.
- Kiểm tra và đánh giá tính ổn định, độ trễ và chính xác của hệ thống điều khiển khi sử dụng ứng dụng Android để điều khiển xe robot.

### ❖ So sánh kết quả đạt được của hệ thống với mục tiêu đã đề ra:

- Điều chỉnh và hiệu chỉnh:

Kiểm tra và hiệu chỉnh các thông số và tín hiệu điều khiển để đảm bảo rằng xe robot di chuyển và quay đúng theo các tín hiệu điều khiển từ ứng dụng Android. Điều chỉnh các tham số như tốc độ, góc quay, và độ nhạy của servo motor và động cơ để đạt hiệu suất tốt nhất.

- Thử nghiệm chức năng:

Thử nghiệm các chức năng của xe robot bằng cách sử dụng ứng dụng Android để điều khiển. Kiểm tra xem xe robot có thể di chuyển theo hướng và tốc độ mong muốn, và liệu servo motor có thể điều chỉnh góc quay chính xác hay không.

- Đánh giá hiệu suất:

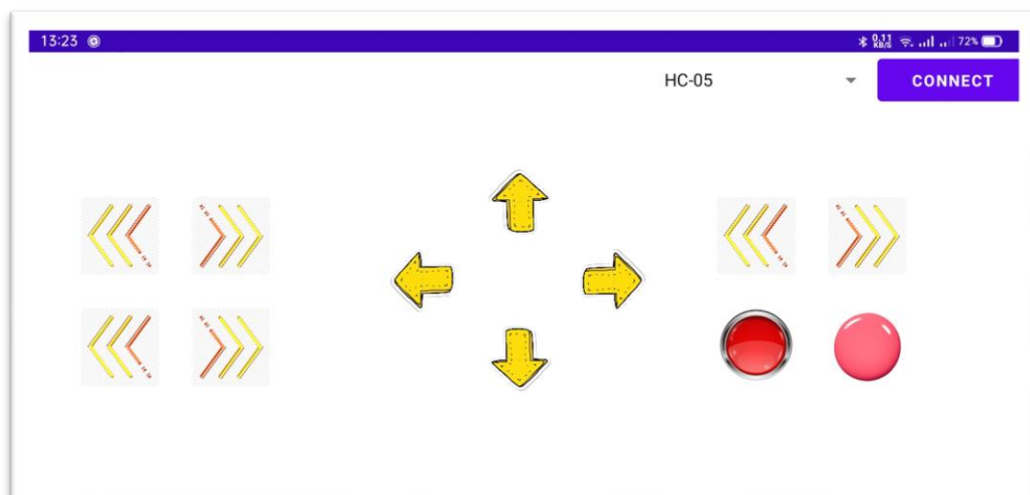
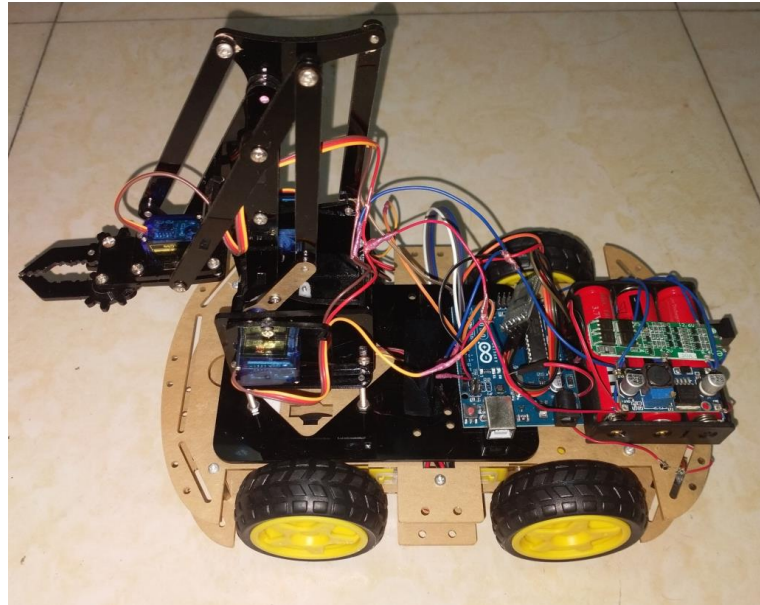
Đánh giá hiệu suất của hệ thống điều khiển và ứng dụng Android. Kiểm tra tính ổn định, độ trễ và độ chính xác của tín hiệu điều khiển từ ứng dụng Android đến Arduino và từ Arduino đến các thành phần điều khiển xe robot.

Đánh giá khả năng phản hồi và độ mượt mà trong quá trình điều khiển. Đảm bảo rằng xe robot có thể phản ứng nhanh chóng và di chuyển một cách trơn tru và chính xác.

- Điều chỉnh và cải tiến:

Dựa trên kết quả kiểm thử và đánh giá, điều chỉnh và cải tiến hệ thống điều khiển và ứng dụng Android nếu cần thiết. Cải tiến có thể bao gồm việc tối ưu hóa mã điều khiển, điều chỉnh thông số hoặc sử dụng các thuật toán tốt hơn để đạt hiệu suất tốt hơn.

#### 4.3. Hướng dẫn vận hành hệ thống

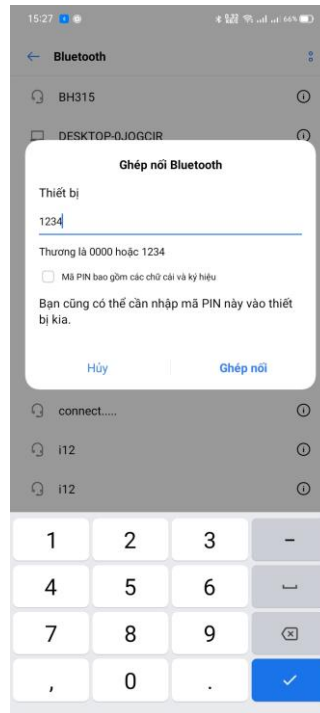


**Hình 4. 1:** Vận hành hệ thống

- Bước 1: Bật công tắc nguồn điện của xe.
- Bước 2: Bật bluetooth trên điện thoại và kết nối với HC05.

Bật bluetooth của điện thoại lên. Sau đó chọn thiết bị kết nối là HC05. Nhập mật khẩu để kết nối với HC05 là 1234.





- Bước 3: Mở app điều khiển android đã được lập trình là tải xuống điện thoại. Chọn select → chọn tên thiết bị muốn kết nối là HC05. Sau đó nhấn connect. Nút connect chuyển thành nút disconnect là đã kết nối thành công.



- Khu vực 4 mũi tên vàng là điều khiển hướng đi của xe. Các mũi tên 2 bên là điều khiển của cánh tay robot.

## KẾT LUẬN

### Kết quả đạt được của đề tài

- Ôn tập và ứng dụng kiến thức của lập trình vi điều khiển, điện tử.
- Nắm được nhiều kỹ thuật trong việc thiết kế mạch in.
- Tìm hiểu tổng quan về hệ thống căn phòng thông minh, thiết kế chế tạo mô hình căn phòng thông minh thành công.

### Hạn chế của đề tài

- Do hạn chế về thời gian và kiến thức nên chương trình code chưa tối ưu.
- Kinh nghiệm còn hạn chế nên hệ thống được thiết kế chưa thực sự tối ưu.
- Hệ thống mới thực hiện được một số chức năng cơ bản của một căn phòng thông minh.

### Hướng phát triển của đề tài

Trong thời gian tới nếu có cơ hội sẽ phát triển thêm những vấn đề chưa làm được như đã nói ở trên và khắc phục những tồn tại của đề tài, hy vọng có thể ứng dụng công nghệ trợ thông minh vào trong nhiều lĩnh vực và tích hợp cho nó thêm nhiều chức năng, ứng dụng rộng rãi ngay trong đời sống hàng ngày chứ không đơn thuần chỉ là trên mô hình.

## TÀI LIỆU THAM KHẢO

- [1] Phạm Ngọc Hưng (2020), “*Đề cương bài giảng môn thiết kế hệ thống nhúng*,” Trường Đại học Sư phạm Kỹ thuật Hưng Yên.
- [2] Đặng Hoài Bắc, Nguyễn Ngọc Minh (2013), Thiết kế hệ tổng nhúng, NXB thông tin và truyền thông.
- [3] Khoa Công nghệ thông tin, “*Đề cương bài giảng môn Lập trình vi điều khiển*”, Trường Đại học Sư phạm Kỹ thuật Hưng Yên.