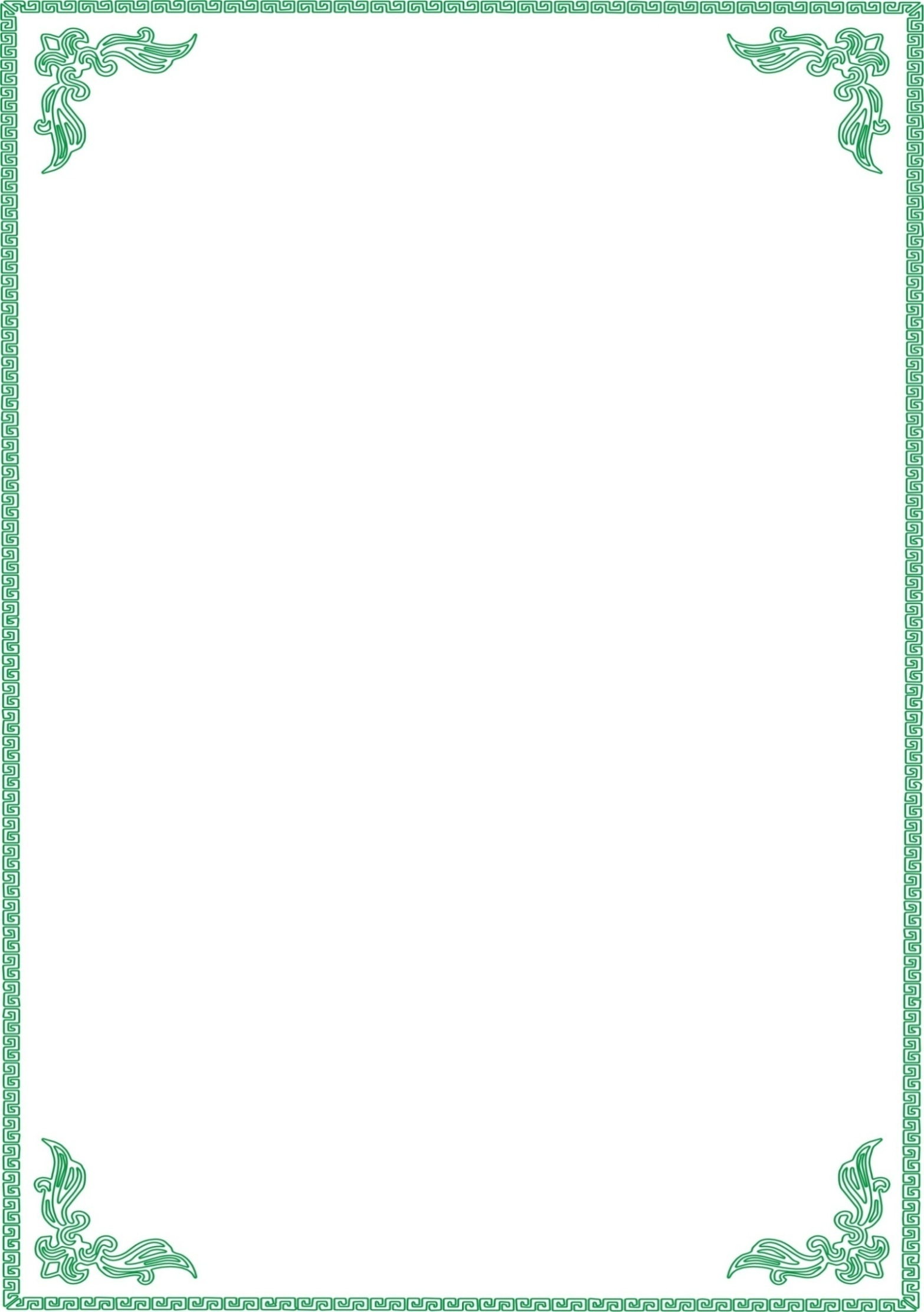
****

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------- ® ----------**

*** ĐỒ ÁN CHUYÊN NGÀNH***

***ĐỀ TÀI:* NAVIGATOR CHO XE MÁY/ĐẠP**

GVHD: **Ths. NGUYỄN QUỐC ĐÍNH**

SVTH: VÕ TẤN VƯƠNG - 11319291

LÊ NGUYỄN HỮU NHÂN - 11279101

LỚP: ĐHTH5TLT

TP. HCM, Tháng 12 Năm 2014

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Phần đánh giá:

Thái độ làm việc:

Kỹ năng làm việc:

Trình bày:

Điểm số: (bằng số) (bằng chữ)

TP. HCM, Ngày… Tháng… Năm…

Giáo viên hướng dẫn

Ths. Nguyễn Quốc Đính

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Phần đánh giá:

Thái độ làm việc:

Kỹ năng làm việc:

Trình bày:

Điểm số: (bằng số) (bằng chữ)

TP. HCM, Ngày… Tháng… Năm…

Giáo viên phản biện

………………………………..

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 - Arduino Nano v3 2](#_Toc405208723)

[Hình 1.2 - Sơ đồ nguyên lý led ma trận 8x8 3](#_Toc405208724)

[Hình 1.3 - Sơ đồ nguyên lý giao tiếp rung 3](#_Toc405208725)

[Hình 2.1 - Kiến trúc Android 5](#_Toc405208735)

[Hình 2.2 - Ứng dụng N4B 7](#_Toc405208736)

[Hình 2.3 - Tab bản đồ 8](#_Toc405208737)

[Hình 2.4 - Tab dẫn đường 9](#_Toc405208738)

[Hình 2.5 - Giao diện IDE Arduino 21](#_Toc405208739)

[Hình 2.6 - Chu trình hoạt động Arduino 22](#_Toc405208740)

[Hình 2.7 - Sơ đồ khối led ma trận 8x8 28](#_Toc405208741)

[Hình 2.8 - Sơ đồ khối giao tiếp rung 29](#_Toc405208742)

**MỤC LỤC**

[CHƯƠNG I. TỔNG QUAN HỆ THỐNG 1](#_Toc405300370)

[1.1. Giới thiệu 1](#_Toc405300371)

[1.2. Mục tiêu 1](#_Toc405300372)

[1.3. Phương pháp thực hiện 1](#_Toc405300373)

[1.3.1. Mạch Arduino 1](#_Toc405300374)

[1.3.2. Ứng dụng 2](#_Toc405300375)

[1.3.2.1. Ứng dụng trong tin học 2](#_Toc405300376)

[1.3.2.2. Ứng dụng trong thực tế 2](#_Toc405300377)

[1.3.3. Mô hình 2](#_Toc405300378)

[CHƯƠNG II. THIẾT KẾ HỆ THỐNG 4](#_Toc405300379)

[2.1. Trên Android 4](#_Toc405300380)

[2.1.1. Tổng quan 4](#_Toc405300381)

[2.1.2. Các chức năng chính của Android 4](#_Toc405300382)

[2.1.3. Kiến trúc của Android 4](#_Toc405300383)

[2.1.4. Cấu trúc một dự án Android 6](#_Toc405300384)

[2.1.5. Chương trình thực thi 6](#_Toc405300385)

[2.2. Trên Arduino 18](#_Toc405300386)

[2.2.1. Tổng quan 18](#_Toc405300387)

[2.2.2. Các chức năng chính 19](#_Toc405300388)

[2.2.3. Môi trường lập trình 20](#_Toc405300389)

[2.2.4. Cơ chế hoạt động 21](#_Toc405300390)

[2.2.5. Chương trình thực thi 22](#_Toc405300391)

[2.2.5.1. Led ma trận 8x8 22](#_Toc405300392)

[2.2.5.2. Giao tiếp rung 27](#_Toc405300393)

[2.2.6. HC-06 bluetooth module 27](#_Toc405300394)

[2.2.7. Thông điệp trao đổi giữa điện thoại và Arduino 28](#_Toc405300395)

[CHƯƠNG III. HƯỚNG PHÁT TRIỂN 30](#_Toc405300396)

[CHƯƠNG IV. TÀI LIỆU THAM KHẢO 31](#_Toc405300397)

**LỜI MỞ ĐẦU**

Ngày nay, thế giới thông tin ngày càng phát triển một cách đa dạng và phong phú. Nhu cầu về thông tin liên lạc trong cuộc sống càng tăng cả về số lượng và chất lượng, đòi hỏi các dịch vụ của ngành viễn thông càng mở rộng. Trong những năm gần đây thông tin vệ tinh trên thế giới đã có những bước tiến vượt bậc đáp ứng nhu cầu đời sống, đưa con người nhanh chóng tiếp cận với các tiến bộ khoa học kỹ thuật.

Nhằm đáp ứng cho các mục đích dẫn đường cũng như xác định vị trí một cách chính xác, nhanh chóng và thuận tiện. Những thiết bị điện thoại di động thì rất bất tiện trong việc quan sát, theo dõi đường xá trong lúc điều khiển xe máy/đạp và câu hỏi đặt ra là cần phải tối ưu hóa vấn đề này đến mức tối ưu nhất?

Ðể tối ưu hóa vấn đề này chúng em chọn đề tài “NAVIGATOR CHO XE MÁY/ĐẠP” làm đồ án chuyên ngành. Đồ án đi sâu thiết kế bản đồ dựa trên cơ sở hệ thống của Google và tạo ra một thiết bị mang tính chất di động nhằm phục vụ cho sự giao tiếp với người sử dụng một cách tự động hóa.

Nội dung của đồ án bao gồm 4 chương:

* CHƯƠNG 1: TỔNG QUAN HỆ THỐNG
* CHƯƠNG 2: THIẾT KẾ HỆ THỐNG
* CHƯƠNG 3: HƯỚNG PHÁT TRIỂN
* CHƯƠNG 4: TÀI LIỆU THAM KHẢO

Với thời gian có hạn cũng như là hạn chế về tài liệu, vì tài liệu về lĩnh vực điện tử không được tiếp cận thường xuyên. Tuy nhiên, bằng nỗ lực bản thân, chứng em đã đáp ứng được yêu cầu của đồ án đề ra, mặc dù không thể không có những thiếu sót. Rất mong được sự đóng góp ý kiến của thầy cô và các bạn để đồ án được hoàn thiện hơn.

TP. HCM, Tháng 12 Năm 2014

Sinh viên thực hiện

Võ Tấn Vương

Lê Nguyễn Hữu Nhân

# CHƯƠNG I TỔNG QUAN HỆ THỐNG

1. **Giới thiệu**

Là hệ thống hiển thị đường đi, có thể giúp nguời dùng giao tiếp được tốt hơn. Định vị trên điện thoại thông minh chắc chắn có thể giúp người đi xe máy hay xe đạp tìm được đường xung quanh thành phố, nhưng không phải ai cũng mạo hiểm gắn điện thoại quý giá của họ ở phía trước tay lái, nên nhóm chúng em đã phát triển dùng hệ thống đèn led với sự trợ giúp của điện thoại.

Những thông tin mà đèn led có thể hiển thị được là:

* Hướng rẽ
* Hướng quay đầu
* Khoảng cách đến điểm rẽ
* Khoảng cách đến đích
* Tốc độ của xe và tốc độ mong muốn theo thời gian thực
* Và nhiều thông tin khác...

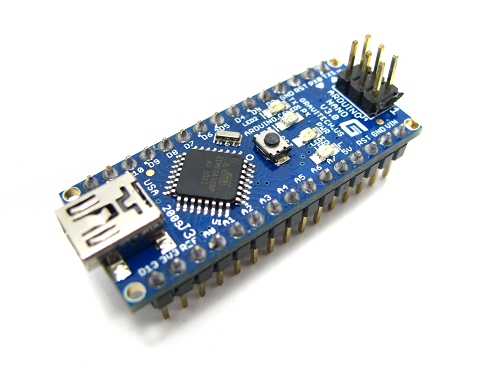
Đèn hiển thị trên màn hình led, mạch bluetooth kết nối với các thiết bị thông minh dùng hệ điều hành Android để định đường đi một cách tốt nhất.

1. **Mục tiêu**

Tương tự như chương trình navigator trên xe hơi, nhóm chúng em sẽ xây dựng chương trình giúp người đi xe máy, xe đạp đi trên con đường đã vạch ra với sự trợ giúp của điện thoại. Hệ thống giao tiếp với người dùng qua thiết bị chỉ dẫn nhỏ bằng led ma trận giao tiếp qua bluetooth.

1. **Phương pháp thực hiện**
2. **Mạch Arduino**

Arduino đã và đang được sử dụng rất rộng rãi trên thế giới, ngày càng chứng tỏ được sức mạnh của chúng thông qua vô số ứng dụng độc đáo của người dùng trong cộng đồng nguồn mở (open-source). Tuy nhiên tại Việt Nam, Arduino vẫn còn chưa được biết đến nhiều.



Hình 1.1 - Arduino Nano v3

1. **Ứng dụng**
2. **Ứng dụng trong tin học**

Giúp sinh viên có nhiều ứng dụng thực tế trong thực tiễn đời sống.

Tập viết viết chương trình trên điện thoại, máy tính bảng chạy hệ điều hành Android.

Cùng với những công việc liên quan đến thiết kế mạch bên khoa điện tử. Nó giúp em rất hứng thú khi khiển khai.

Đồ án này giúp ích nhiều cho công việc sau này của chúng em. Nhiều ứng dụng rất hay về điện tử và tin học mà nhóm chúng em được tiếp xúc.

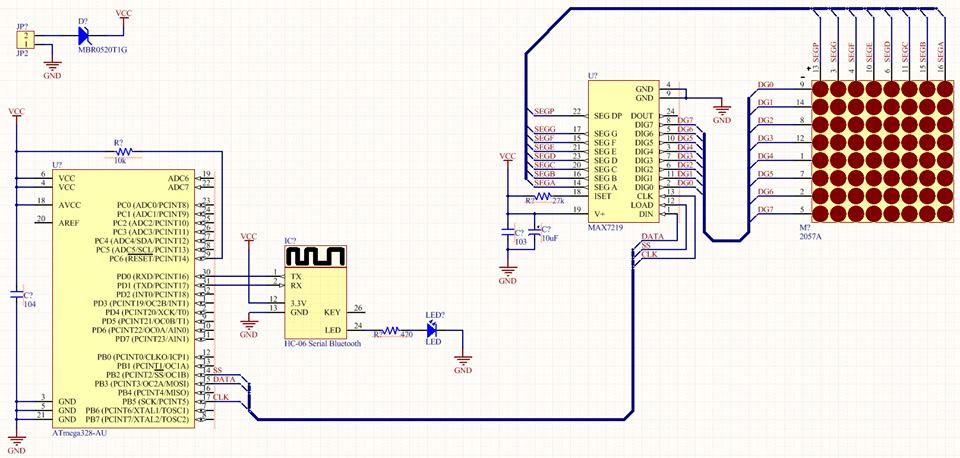
1. **Ứng dụng trong thực tế**

Giúp người đi đường có thể phán đoán dễ dàng lộ trình đường đi.

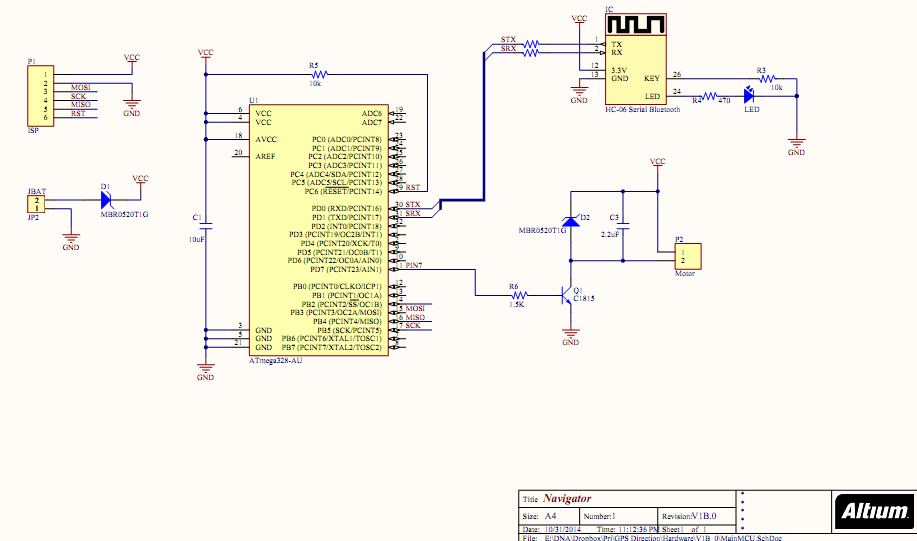
Giúp đỡ tốt cho những người không quen thuộc đường xá.

Giúp giao tiếp tốt và tạo sự dễ dàng nhất cho người sử dụng.

1. **Mô hình**

****

Hình 1.2 - Sơ đồ nguyên lý led ma trận 8x8



Hình 1.3 - Sơ đồ nguyên lý giao tiếp rung

# CHƯƠNG II THIẾT KẾ HỆ THỐNG

1. **Trên Android**
2. **Tổng quan**

Android là một hệ điều hành dành cho thiết bị di động như điện thoại, máy tính bảng và netbooks. Android được phát triển bởi Google, dựa trên nền tảng Linux kernel và các phần mềm nguồn mở. Ban đầu nó được phát triển bởi Android Inc (sau đó được Google mua lại) và gần đây nó trở thành một trong những phần mềm đứng đầu của liên minh OHA (Open Handset Alliance - với khoảng 78 thành viên bao gồm cả nhà sản xuất, nhà phát triển ứng dụng... cho thiết bị di dộng mà dẫn đầu là Google).

Android có một cộng đồng phát triển ứng dụng rất lớn, hiện có khoảng hơn 70.000 ứng dụng có sẵn cho Android và đang liên tục được cập nhật. Ứng dụng được phát triển bằng ngôn ngữ Java kết hợp với thư viện Java có sẵn của Google. Các nhà phát triển ứng dụng có thể sử dụng máy tính chạy hệ điều hành Windows hoặc MacOS hoặc Linux kết hợp với Android SDK để phát triển ứng dụng cho Android. Hệ điều hành Android bao gồm 12.000.000 dòng mã trong đó có 3.000.000 dòng XML, 2.800.000 dòng C, 2.100.000 dòng Java, và 1.750.000 dòng C++.

Để phát triển phần mềm trên Android, các lập trình viên có thể tải về bộ công cụ phát triển (Android SDK). Bộ công cụ bao gồm các công cụ và các API cần thiết, sử dụng ngôn ngữ Java để lập trình.

1. **Các chức năng chính của Android**

* Android cung cấp framework ứng dụng cho phép việc tái sử dụng và thay thế mã nguồn ở dạng component một cách dễ dàng.
* Cung cấp máy ảo Dalvik được tối ưu cho các thiết bị di động.
* Trình duyệt Web dựa trên engine mã nguồn mở Webkit.
* Các tính năng đồ họa được tối ưu bởi một thư viện đồ họa 2D bên dưới; đối với đồ họa 3D, Android sử dụng thư viện OpenGL ES 1.0 nếu thiết bị có hỗ trợ.
* Sử dụng SQLite để lưu trữ dữ liệu có cấu trúc.
* Hỗ trợ các định dạng hình ảnh, âm thanh, video phổ biến như MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF.
* Hỗ trợ băng tầng GSM (tùy vào phần cứng thiết bị).
* Hỗ trợ Bluetooth, EDGE, 3G, WiFi (tùy vào phần cứng thiết bị).
* Ngoài ra còn có khả năng của các thiết bị như máy chụp ảnh, thiết bị định vị toàn cầu, la bàn, và bộ cảm biến gia tốc.
* Cung cấp môi trường phát triển phần mềm đầy đủ, bao gồm phần mềm giả lập thiết bị, các công cụ gỡ rối (debugging), theo dõi bộ nhớ và năng suất hoạt động, plugin cho Eclipse IDE.

1. **Kiến trúc của Android**

****

Hình 2.1 - Kiến trúc Android

**Tầng ứng dụng – Applications**

Tầng ứng dụng của Android bao gồm các ứng dụng cốt lõi như:

* Email client
* Chương trình SMS
* Lịch
* Bản đồ
* Trình duyệt
* Danh bạ
* Các ứng dụng khác.

**Tầng Application Framework**

Ở tầng này, các nhà phát triển ứng dụng có thể truy cập vào phần cứng thiết bị, thông tin định vị vị trí, chạy các dịch vụ nền, đặt các cảnh báo, thông báo vào thanh trạng thái, v.v…. và quan trọng nhất, đó là các API của framework.

Phía dưới tất cả các ứng dụng là một tập các dịch vụ hệ thống bao gồm:

* Một tập các đối tượng View có thể được mở rộng để xây dựng một ứng dụng, gồm có List, Grid, TextBox, Button, và WebBrowser.
* Các đối tượng ContentProvider cho phép các ứng dụng truy cập vào dữ liệu của các ứng dụng khác (chẳng hạn như truy cập danh bạ), hoặc để chia sẽ dữ liệu với nhau.
* Một trình quản lý tài nguyên, cho phép truy xuất các tài nguyên không phải mã nguồn như các chuỗi đã được bản địa hóa, các tập tin đồ họa và giao diện.
* Trình quản lý thông báo cho phép tất cả các ứng dụng hiển thị các cảnh báo lên thanh trạng thái.
* Trình quản lý các đối tượng Activity dùng để quản lý vòng đời của các ứng dụng cung cấp các chức năng điều hướng.

**Tầng Libraries – Runtime**

Tầng này cung cấp các thư viện media dựa trên thư viện PacketVideo’s OpenCORE; các thư viện này hỗ trợ khả năng playback và thu lại nhiều định dạng âm thanh, hình ảnh thông dụng như MPEG4, H.264, MP3, AAC, AMR, JPG, PNG.

Kèm theo đó là thư viện SQLite, một hệ quản trị cơ sở dữ liệu nhỏ nhẹ và mạnh mẽ được cung cấp cho tất cả các ứng dụng.

Ở Runtime, Android cung cấp máy ảo Dalvik dùng để thực thi các file định dạng Dalvik Executable (.dex) đã được tối ưu hóa cho các thiết bị có bộ nhớ nhỏ. Máy ảo Dalvik chỉ chạy các class đã được đăng ký và được biên dịch bởi một trình biên dịch Java đi kèm theo bộ SDK (dx tool). Ngoài ra Dalvik còn sử dụng nhân Linux để quản lý các tính năng ở mức thấp và các tác vụ chạy theo luồng. Mọi ứng dụng Android chạy trên một tiến trình riêng cùng với một instance riêng của máy ảo Dalvik.Dalvik đã được tối ưu sao cho một thiết bị có thể chạy nhiều mày ảo cùng lúc một cách hiệu quả.

**Tầng Linux kernel**

Android được phát triển dựa trên các dịch vụ hệ thống cốt lõi của Linux phiên bản 2.6, bao gồm các module:

* Security
* Memory management
* Process management
* Network stack
* Driver model

Tầng kernel hoạt động như một lớp trừu tượng giữa lớp phần cứng và phần mềm.

1. **Cấu trúc một dự án Android**

* AndroidManifest.xml, là một file XML miêu tả ứng dụng được xây dựng và các thành phần kèm theo như các Activities, các Services được cung cấp bởi ứng dụng đó.
* Thư mục libs/ chứa các thư viện Java ở dạng JAR của các hãng thứ 3 mà ứng dụng cần để chạy.
* Thư mục src/ chứa mã nguồn Java cho ứng dụng.
* Thư mục res/ chứa các tài nguyên chẳng hạn như các biểu tượng, giao diện, … được đóng gói kèm theo khi biên dịch ứng dụng.
* Thư mục assets/ chứa các file tĩnh dùng cài đặt lên hệ thống.

1. **Chương trình thực thi**

Wifi/3G/GPRS/… kết nối mạng lấy dữ liệu bản đồ về và xử lý.

GPS giúp xác định vị trí hiện tại. nguồn

Bluetooth kết nối đến Arduino(AVR) để điều khiển tín hiệu đèn.



Hình 2.2 - Ứng dụng N4B

Key đăng ký từ dịch vụ phát triển của Google Maps

<meta-data

android:name=*"com.google.android.maps.v2.API\_KEY*  android:value=*"AIzaSyChO79Ok6ATVEsV2mHU-bdAH6yNGTpPpmw"*/>

Xử lý quẹo trái, phải

**for**(**int** i=0;i<listInstructions.size();i++) //vòng lặp chạy hết tất cả điểm nút

**if**(listInstructions.get(i).contains("Rẽ trái") || listInstructions.get(i).contains("Chếch sang trái")) //xử lý điểm nút quẹo trái

{

latitude[pos] = (**double**)listlatlng.get(i).latitude;

longitude[pos] = (**double**)listlatlng.get(i).longitude;

direction[pos] = "left";

pos++;

}

**else** **if**(listInstructions.get(i).contains("Rẽ phải") || listInstructions.get(i).contains("Chếch sang phải")) //xử lý điểm nút quẹo phải

{

latitude[pos] = (**double**)listlatlng.get(i).latitude;

longitude[pos] = (**double**)listlatlng.get(i).longitude;

direction[pos] = "right";

pos++;

}

Xử lý khoảng cách

**public** **int** Distances(**double** lat, **double** lng, **float** dsc)

{

**float**[] distance = **new** **float**[2];

Location.distanceBetween(ltBeg, lnBeg, lat, lng, distance); //tính khoảng cách vĩ độ, kinh độ của điểm bắt đầu và điểm đích để trả về kết quả ở distance

**if**(distance[0] <= dsc) //so sánh kết quả

{

latGoal=0;

lngGoal=0;

**return** 0;

}

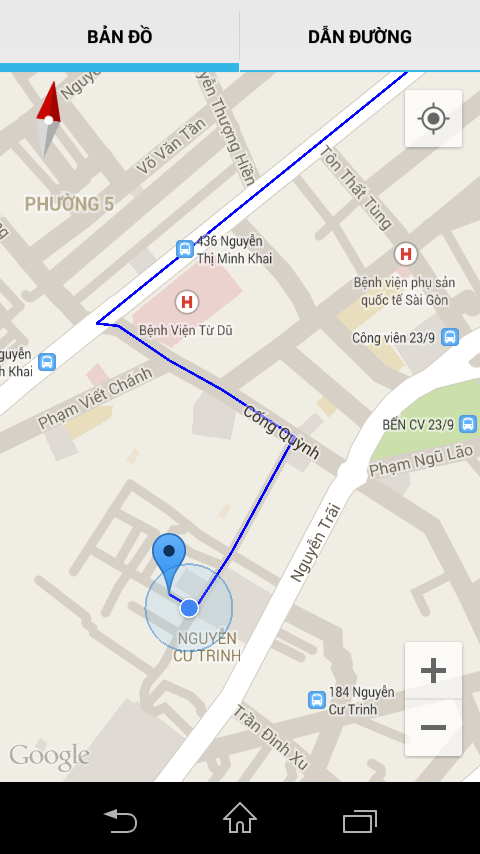
**return** 1;

}

Chạm một điểm cần đến trên màn hình trong tab bản đồ để bắt đầu lộ trình.

Nhấn dài trên màn hình để xóa lộ trình đã vạch ra.

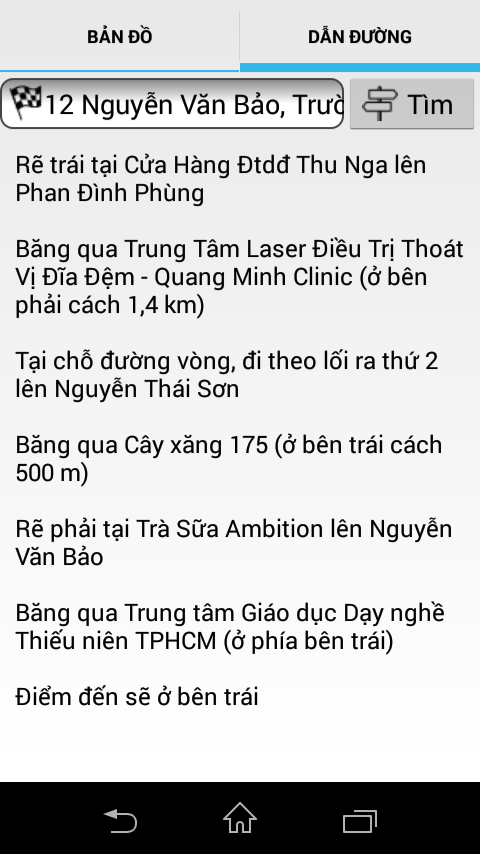
Nhằm mục đích tạo sự thuận lợi nhất cho người dùng, chương trình đã được tối ưu hóa chỉ trong khoảng thời gian ngắn hai đến ba giây người dùng có thể bắt đầu lộ trình muốn đến.



Hình 2.3 - Tab bản đồ

Giúp người dùng tìm kiếm đường đi đến nơi cần tìm một cách nhanh chóng.

Hiển thị chi tiết về tất cả lộ trình sẽ đến theo dạng danh sách.



Hình 2.4 - Tab dẫn đường

Gởi dữ liệu qua bluetooth

**private** **void** sendData(String message)

{

**byte**[] msgBuffer = message.getBytes(); //khai báo và đổi giá trị chuỗi ra byte

Log.*d*(*TAG*, "...Send data: " + message + "..."); //lưu vào file log trên điện thoại

**try**

{

outStream.write(msgBuffer); //gởi dữ liệu qua bluetooth

}

**catch** (IOException e)

{

**if** (*address*.equals("00:00:00:00:00:00"))

Toast.*makeText*(getBaseContext(), "Fatal Error", Toast.*LENGTH\_SHORT*).show(); //xuất ra thông báo lỗi trên màn hình điện thoại khi không kết nối được với bluetooth

}

}

Lấy vị trí đầu tiên

**private** **void** getLocation()

{

googleMap.clear();

markerPoints.clear();

LatLng latLng = **new** LatLng(ltBeg, lnBeg);

onMarker(latLng, **null**);

googleMap.animateCamera(CameraUpdateFactory.*newLatLng*(latLng));

position=**true**;

}

Cập nhật vị trí GPS hiện tại liên tục mỗi giây để tạo ra sự chính xác cho việc dẫn đường đến thiết bị Arduino

locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*, 400, 0, onLocationChange); //cập nhật vị trí hiện tại mỗi 400 mili giây

Xử lý vị trí GPS hiện tại, đưa ra kết quả đã xử lý đến thiết bị và thông báo cho người dùng quẹo trái hay phải trong khoảng cách 100 mét

* Giao tiếp đèn led

**public** **void** onLocationChanged(Location location)

{

**try**

{

ltBeg = location.getLatitude(); //lấy vị trí vĩ độ hiện tại

lnBeg = location.getLongitude(); //lấy vị trí kinh độ hiện tại

**if**(position == **false**)

{

getLocation();

sendData("3");

}

**if**(markerPoints.size() > 1)

{

**if**(pos < latitude.length && Distances(latitude[pos], longitude[pos], 100) == 0)

{

**if**(direction[pos].equals("left"))

{

sendData("0"); //xử lý quẹo trái

Thread.*sleep*(20000);

pos++;

}

**else** **if**(direction[pos].equals("right"))

{

sendData("1"); //xử lý quẹo phải

Thread.*sleep*(20000);

pos++;

}

}

sendData("2"); //xử lý đi thẳng

}

}

**catch** (Exception e)

{

e.printStackTrace();

}

}

* Giao tiếp rung

**public** **void** onLocationChanged(Location location)

{

**try**

{

ltBeg = location.getLatitude(); //lấy vị trí vĩ độ hiện tại

lnBeg = location.getLongitude(); //lấy vị trí kinh độ hiện tại

**if**(position == **false**)

getLocation();

**if**(markerPoints.size() > 1)

{

**if**(pos < latitude.length && Distances(latitude[pos], longitude[pos], 100) == 0)

{

**if**(direction[pos].equals("left"))

{

**for**(**int** i=0;i<3;i++)

{

sendData("0"); //xử lý quẹo trái

Thread.*sleep*(1000);

}

pos++;

}

**else** **if**(direction[pos].equals("right"))

{

**for**(**int** i=0;i<3;i++)

{

Vibrator v = (Vibrator) getSystemService(Context.*VIBRATOR\_SERVICE*); //xử lý quẹo phải

v.vibrate(4000);

Thread.*sleep*(5000);

}

pos++;

}

}

}

}

**catch** (Exception e)

{

Toast.*makeText*(MainActivity.**this**, e.toString(), Toast.*LENGTH\_LONG*).show(); //xuất ra thông báo lỗi trên màn hình điện thoại khi có tình huống lỗi xảy ra

}

}

Loại bỏ chế độ chạy ngầm khi người dùng muốn thoát

@Override

**public** **void** onDestroy()

{

android.os.Process.*killProcess*(android.os.Process.*myPid*()); //xử lý tắt chạy ngầm

**super**.onDestroy();

}

Bật Wifi

**private** **void** onWifi()

{

**try**

{

WifiManager wifiManager = (WifiManager)**this**.getSystemService(MainActivity.*WIFI\_SERVICE*);

wifiManager.setWifiEnabled(**true**);

}

**catch** (Exception e)

{

e.printStackTrace();

}

}

Bật dữ liệu di động (3G/GPRS)

ConnectivityManager manager;

manager = (ConnectivityManager) getSystemService(*CONNECTIVITY\_SERVICE*);

**private** **void** onMobileData()

{

**try**

{

Method dataMtd = **null**;

dataMtd = ConnectivityManager.**class**.getDeclaredMethod("setMobileDataEnabled", **boolean**.**class**);

dataMtd.setAccessible(**true**);

dataMtd.invoke(manager, **true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

Kiểm tra trạng thái mạng và hiện thông báo chọn lựa kết nối

**private** **void** checkNetworkState()

{

**try**

{

wifi = manager.getNetworkInfo(ConnectivityManager.*TYPE\_WIFI*).isConnectedOrConnecting(); //kiểm tra kết nối wifi

mobile = manager.getNetworkInfo(ConnectivityManager.*TYPE\_MOBILE*).isConnectedOrConnecting(); //kiểm tra kết nối dữ liệu di động

**if** (!wifi && !mobile) //nếu cả 2 wifi và dữ liệu di động không có kết nối thì sẽ hiện hộp thoại yêu cầu kết nối

{

AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(**this**);

alertDialogBuilder.setMessage("Kết nối mạng chưa được bật").setCancelable(**false**).setPositiveButton("Wifi", **new** DialogInterface.OnClickListener() //nút bật wifi

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

onWifi(); //thực thi hàm bật wifi

Toast.*makeText*(getBaseContext(), "Wifi đã được bật", Toast.*LENGTH\_LONG*).show();

}

});

alertDialogBuilder.setNeutralButton("3G/GPRS", **new** DialogInterface.OnClickListener() //nút bật 3G/GPRS

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

onMobileData(); //thực thi hàm bật dữ liệu di động

Toast.*makeText*(getBaseContext(), "Dữ liệu di động đã được bật", Toast.*LENGTH\_LONG*).show();

}

});

alertDialogBuilder.setNegativeButton("Hủy", **new** DialogInterface.OnClickListener() //nút hủy

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

dialog.cancel();

}

});

AlertDialog alert = alertDialogBuilder.create();

alert.show();

}

}

**catch** (Exception e)

{

e.printStackTrace();

}

}

Kết nối đến bluetooth trên thiết bị dẫn đường

**private** **boolean** connectBT()

{

**try**

{

device = btAdapter.getRemoteDevice(*address*); //điều hướng đến địa chỉ MAC của bluetooth trên thiết bị

btSocket = createBluetoothSocket(device); //tạo socket kết nối

btAdapter.cancelDiscovery(); //hủy chức năng khám phá bluetooth

btSocket.connect(); //kết nối đến thiết bị

outStream = btSocket.getOutputStream(); //tạo luồng dữ liệu xuất

Vibrator v = (Vibrator)getSystemService(Context.*VIBRATOR\_SERVICE*);

v.vibrate(300); //báo rung 300 mili giây khi đến nối thành công

sendData("3"); //gởi dữ liệu đến thiết bị đã đến kết nối thành công

**return** **true**; //trả về trạng thái kết nối thành công

}

**catch** (Exception e)

{

**return** **false**; //trả về trạng thái lỗi nếu không kết nối được thiết bị

}

}

Ghép đến bluetooth của thiết bị

**private** **void** pairBT()

{

**try**

{

**if** (device.getBondState() == BluetoothDevice.*BOND\_BONDED*)

{

method = device.getClass().getMethod("removeBond", (Class[]) **null**);

method.invoke(device, (Object[]) **null**);

method = device.getClass().getMethod("createBond", (Class[]) **null**);

method.invoke(device, (Object[]) **null**);

connectBT();

}

**else**

{

method = device.getClass().getMethod("createBond", (Class[]) **null**);

method.invoke(device, (Object[]) **null**);

connectBT();

}

}

**catch** (Exception e)

{

Toast.*makeText*(getBaseContext(), "Vui lòng cài đặt ghép nối đến thiết bị dẫn đường", Toast.*LENGTH\_LONG*).show();

}

}

Kiểm tra trạng thái bluetooth và hiện thông báo chọn lựa kết nối

**private** **void** checkBTState()

{

**try**

{

**if**(btAdapter == **null**)

{

Toast.*makeText*(getBaseContext(), "Bluetooth không hỗ trợ", Toast.*LENGTH\_SHORT*).show();

}

**else**

{

**if** (!btAdapter.isEnabled()) //bluetooth chưa được bật

{

AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(**this**);

alertDialogBuilder.setMessage("Bluetooth chưa được bật để kết nối đến thiết bị dẫn đường").setCancelable(**false**).setPositiveButton("Bật", **new** DialogInterface.OnClickListener() //nút bật

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

btAdapter.enable(); //bật bluetooth

Toast.*makeText*(getBaseContext(), "Bluetooth đã được bật", Toast.*LENGTH\_LONG*).show();

checkConnection("Đang kiểm tra kết nối đến thiết bị dẫn đường", 1);

}

});

alertDialogBuilder.setNegativeButton("Hủy", **new** DialogInterface.OnClickListener() //nút hủy

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

dialog.cancel();

}

});

AlertDialog alert = alertDialogBuilder.create();

alert.show();

}

**else** //bluetooth đã được bật

connectBT(); //gọi lại hàm kết nối bluetooth của thiết bị

}

}

**catch** (Exception e)

{

e.printStackTrace();

}

}

Kiểm tra trạng thái GPS và hiện thông báo kết nối

**private** **void** showGPSDisabled()

{

AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(**this**);

alertDialogBuilder.setMessage("GPS chưa hoạt động").setCancelable(**false**).setPositiveButton("Bật", **new** DialogInterface.OnClickListener() //nút bật

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

Intent callGPSSettingIntent = **new** Intent( android.provider.Settings.*ACTION\_LOCATION\_SOURCE\_SETTINGS*);

startActivity(callGPSSettingIntent); //bật chức năng GPS

}

});

alertDialogBuilder.setNegativeButton("Hủy", **new** DialogInterface.OnClickListener()

{

**public** **void** onClick(DialogInterface dialog, **int** id)

{

dialog.cancel();

}

});

AlertDialog alert = alertDialogBuilder.create();

alert.show();

}

Đa ngôn ngữ

lang = locale.getCountry();

**if**(lang.equals("VN"))

lang = "vi";

* Resource Tiếng Anh

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<resources>

<string name=*"app\_name"*>N4B</string>

<string name=*"find"*>Find</string>

<string name=*"hint"*>Enter a place</string>

<string name=*"image"*>Powered by Google</string>

</resources>

* Resource Tiếng Việt

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<resources>

<string name=*"app\_name"*>N4B</string>

<string name=*"find"*>Tìm</string>

<string name=*"hint"*>Nhập địa chỉ cần đến</string>

<string name=*"image"*>Được hỗ trợ bởi Google</string>

</resources>

**private** **void** LoadLanguage() //chuyển sang ngôn ngữ máy

{

**if**(lang.equals("vi"))

{

eb = "Bật";

cc = "Hủy";

bf = "Đang kiểm tra kết nối mạng";

gm = "Kiểm tra kết nối mạng và thử lại";

sl = "Chưa nhập địa chỉ cần đến";

ld = "Đang tiến hành tải dữ liệu";

nw = "Vui lòng cài đặt kết nối mạng";

sb = "Kết nối đến thiết bị dẫn đường thành công";

ns = "Kết nối mạng chưa được bật";

wf = "Wifi đã được bật";

gr = "Dữ liệu di động đã được bật";

bt = "Bluetooth đã được bật";

pb = "Thiết bị dẫn đường chưa hoạt động";

pt = "Vui lòng cài đặt ghép nối đến thiết bị dẫn đường";

bs = "Bluetooth không hỗ trợ";

tb = "Bluetooth chưa được bật để kết nối đến thiết bị dẫn đường";

cb = "Đang kết nối đến thiết bị dẫn đường";

sd = "Không có kết nối đến thiết bị dẫn đường";

to = "BẢN ĐỒ";

tt = "DẪN ĐƯỜNG";

gd = "GPS chưa hoạt động";

lg = "language=vi";

lc = "Nhấn dài trên bản đồ để xóa địa chỉ đích";

wl = "Chờ tìm định vị...";

}

**else**

{

eb = "Enable";

cc = "Cancel";

bf = "Checking network connection";

gm = "Check network connection and try again";

sl = "Please enter the address where you want to go";

ld = "Loading data";

nw = "Please install network connection";

sb = "Your navigation device has connected";

ns = "Network connection is not enabled";

wf = "Wifi is turned on";

gr = "Mobile data is turned on";

bt = "Bluetooth is turned on";

pb = "Your navigation device is not working";

pt = "Please pair your navigation device";

bs = "Bluetooth is not supported";

tb = "Bluetooth is not enabled for connecting to navigation device";

cb = "Connecting to navigation device";

sd = "Your navigation device has not connected";

to = "MAP";

tt = "DIRECTION";

gd = "GPS is not working";

lg = "language=en";

lc = "Long click on map that delete destination address";

wl = "Waiting for location...";

}

}

1. **Trên Arduino**
2. **Tổng quan**

Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm. Chỉ với khoảng $30, người dùng đã có thể sở hữu một bo Arduino có 20 ngõ I/O có thể tương tác và điều khiển thiết bị.

Arduino ra đời tại thị trấn Ivrea thuộc nước Ý và được đặt theo tên một vị vua vào thế kỷ thứ 9 là King Arduin. Arduino chính thức được đưa ra giới thiệu vào năm 2005 như là một công cụ khiêm tốn dành cho các sinh viên của giáo sư Massimo Banzi, là một trong những người phát triển Arduino, tại trường Interaction Design Instistute Ivrea (IDII). Mặc dù hầu như không được tiếp thị gì cả, tin tức về Arduino vẫn lan truyền với tốc độ chóng mặt nhờ những lời truyền miệng tốt đẹp của những người dùng đầu tiên. Hiện nay Arduino nổi tiếng tới nỗi có người tìm đến thị trấn Ivrea chỉ để tham quan nơi đã sản sinh ra Arduino.

1. **Các chức năng chính**

Bo mạch Arduino sử dụng dòng vi xử lý 8-bit megaAVR của Atmel với hai chip phổ biến nhất là ATmega328 và ATmega2560. Các dòng vi xử lý này cho phép lập trình các ứng dụng điều khiển phức tạp do được trang bị cấu hình mạnh với các loại bộ nhớ ROM, RAM và Flash, các ngõ vào ra digital I/O trong đó có nhiều ngõ có khả năng xuất tín hiệu PWM, các ngõ đọc tín hiệu analog và các chuẩn giao tiếp đa dạng như UART, SPI, TWI (I2C).

**Tốc độ xử lý**

* Xung nhịp: 16MHz
* EEPROM: 1KB (ATmega328) và 4KB (ATmega2560)
* SRAM: 2KB (Atmega328) và 8KB (Atmega2560)
* Flash: 32KB (Atmega328) và 256KB (Atmega2560)

**Đọc tín hiệu cảm biến ngõ vào**

Digital:

* Các bo mạch Arduino đều có các cổng digital có thể cấu hình làm ngõ vào hoặc ngõ ra bằng phần mềm. Do đó người dùng có thể linh hoạt quyết định số lượng ngõ vào và ngõ ra.
* Tổng số lượng cổng digital trên các mạch dùng Atmega328 là 14, và trên Atmega2560 là 54.

Analog:

* Các bo mạch Arduino đều có trang bị các ngõ vào analog với độ phân giải 10-bit (1024 phân mức, với điện áp chuẩn là 5V thì độ phân giải khoảng 0.5mV).
* Số lượng cổng vào analog là 6 đối với Atmega328, và 16 đối với Atmega2560.
* Với tính năng đọc analog, người dùng có thể đọc nhiều loại cảm biến như nhiệt độ, áp suất, độ ẩm, ánh sáng, gyro, accelerometer…

**Xuất tín hiệu điều khiển ngõ ra**

Digital:

* Tương tự như các cổng vào digital, người dùng có thể cấu hình trên phần mềm để quyết định dùng ngõ digital nào là ngõ ra.
* Tổng số lượng cổng digital trên các mạch dùng Atmega328 là 14, và trên Atmega2560 là 54.

PWM:

* Trong số các cổng digital, người dùng có thể chọn một số cổng dùng để xuất tín hiệu điều chế xung PWM. Độ phân giải của các tín hiệu PWM này là 8-bit.
* Số lượng cổng PWM đối với các bo dùng Atmega328 là 6, và đối với các bo dùng Atmega2560 là 14.
* PWM có nhiều ứng dụng trong viễn thông, xử lý âm thanh hoặc điều khiển động cơ mà phổ biến nhất là động cơ servos trong các máy bay mô hình.

**Chuẩn giao tiếp**

Serial:

* Đây là chuẩn giao tiếp nối tiếp được dùng rất phổ biến trên các mạch Arduino. Mỗi mạch có trang bị một số cổng Serial cứng (việc giao tiếp do phần cứng trong chip thực hiện). Bên cạnh đó, tất cả các cổng digital còn lại đều có thể thực hiện giao tiếp nối tiếp bằng phần mềm (có thư viện chuẩn, người dùng không cần phải viết code). Mức tín hiệu của các cổng này là TTL 5V.
* Số lượng cổng Serial cứng của Atmega328 là 1 và của Atmega2560 là 4.
* Với tính năng giao tiếp nối tiếp, các mạch Arduino có thể giao tiếp được với rất nhiều thiết bị như PC, touchscreen, các game console…

USB:

* Các mạch Arduino tiêu chuẩn đều có trang bị một cổng USB để thực hiện kết nối với máy tính dùng cho việc tải chương trình. Tuy nhiên các chip AVR không có cổng USB, do đó các mạch Ardunino phải trang bị thêm phần chuyển đổi từ USB thành tín hiệu UART. Do đó máy tính nhận diện cổng USB này là cổng COM chứ không phải là cổng USB tiêu chuẩn.

SPI:

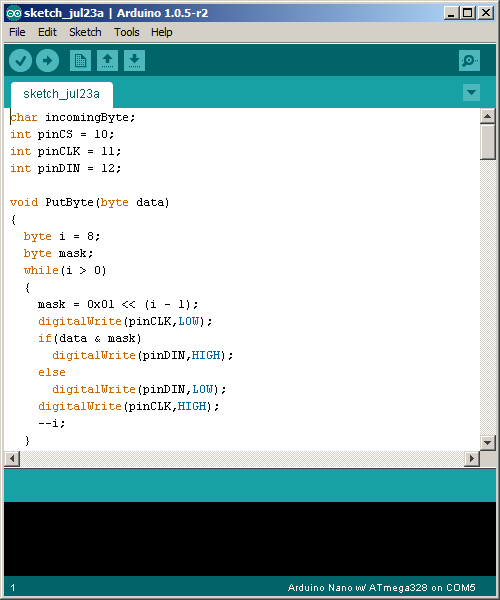
* Đây là một chuẩn giao tiếp nối tiếp đồng bộ có bus gồm có 4 dây. Với tính năng này các mạch Arduino có thể kết nối với các thiết bị như LCD, bộ điều khiển video game, bộ điều khiển cảm biến các loại, đọc thẻ nhớ SD và MMC…

TWI (I2C):

* Đây là một chuẩn giao tiếp đồng bộ khác nhưng bus chỉ có hai dây. Với tính năng này, các bo Arduino có thể giao tiếp với một số loại cảm biến như thermostat của CPU, tốc độ quạt, một số màn hình OLED/LCD, đọc real-time clock, chỉnh âm lượng cho một số loại loa…

1. **Môi trường lập trình**

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

****

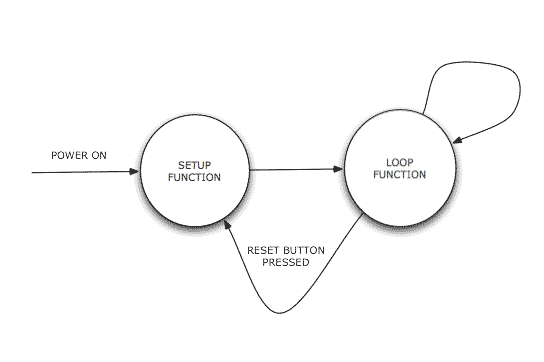
Hình 2.5 - Giao diện IDE Arduino

Môi trường lập trình Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR C vào chương trình nếu muốn.

1. **Cơ chế hoạt động**

Một chương trình Arduino sẽ bắt buộc có hai hàm: setup() và loop(). Khi cắm điện cho Arduino, reset hay nạp chương trình mới cho nó, hàm setup() sẽ được gọi đến đầu tiên. Sau khi xử lý xong hàm setup(), Arduino sẽ nhảy đến hàm loop() và lặp vô hạn hàm đó.



Hình 2.6 - Chu trình hoạt động Arduino

Hàm setup sẽ chứa tất cả các khai báo ban đầu của chương trình (vì nó luôn được chạy đầu tiên và chỉ chạy 1 lần). Hàm loop sẽ là nơi thực hiện các tác vụ, nơi các lệnh được lặp đi lặp lại vô hạn.

1. **Chương trình thực thi**
2. **Led ma trận 8x8**

boolean start; //khai báo true/false bật/tắt đèn quét ma trận

char incomingByte; //khai báo giá trị nhận từ bluetooth

int pinCS = 10; //khai báo pin load

int pinCLK = 11; //khai báo pin clock

int pinDIN = 12; //khai báo pin data

void PutByte(byte data)

{

byte i = 8;

byte mask;

while(i > 0)

{

mask = 0x01 << (i - 1); //dịch bit sang trái theo (i - 1)

digitalWrite(pinCLK,LOW);

if(data & mask) //chọn bit

digitalWrite(pinDIN,HIGH); //vị trí bit ở trạng thái tích cực

else

digitalWrite(pinDIN,LOW); //vị trí bit ở trạng thái không tích cực

digitalWrite(pinCLK,HIGH); //xác nhận đọc dữ liệu vào

--i;

}

}

void MaxSingle(byte reg,byte col)

{

digitalWrite(pinCS,LOW);

PutByte(reg); //dữ liệu thanh ghi

PutByte(col); //dữ liệu led

digitalWrite(pinCS,LOW);

digitalWrite(pinCS,HIGH); //xác nhận tải dữ liệu vào

}

void Init\_MAX7219()

{

MaxSingle(0x09,0x00); //giải mã BCD

MaxSingle(0x0a,0x03); //độ sáng led mức 3

MaxSingle(0x0b,0x07); //khai báo led

MaxSingle(0x0c,0x01); //chế độ normal

MaxSingle(0x0f,0x00); //kiểm tra toàn bộ led

}

void Left()

{

MaxSingle(1,1); //- - - - - - - +

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(2,56); //- - + + + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(3,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(4,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(5,238); //+ + + - + + + -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(6,68); //- + - - - + - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(7,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(8,16); //- - - + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

}

void Right()

{

MaxSingle(1,1); //- - - - - - - +

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(2,16); //- - - + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(3,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(4,68); //- + - - - + - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(5,238); //+ + + - + + + -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(6,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(7,40); //- - + - + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(8,56); //- - + + + - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

}

void Head()

{

MaxSingle(1,1); //- - - - - - - +

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(2,16); //- - - + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(3,48); //- - + + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(4,94); //- + - + + + + -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(5,130); //+ - - - - - + -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(6,94); //- + - + + + + -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(7,48); //- - + + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

MaxSingle(8,16); //- - - + - - - -

MaxSingle(0x0c,0x00);

delay(15);

MaxSingle(0x0c,0x01);

}

void Begin()

{

MaxSingle(8,255); //+ + + + + + + +

delay(100);

MaxSingle(8,0);

MaxSingle(7,255); //+ + + + + + + +

delay(100);

MaxSingle(7,0);

MaxSingle(6,255); //+ + + + + + + +

delay(100);

MaxSingle(6,0);

MaxSingle(5,255); //+ + + + + + + +

delay(100);

MaxSingle(5,0);

MaxSingle(4,255); //+ + + + + + + +

delay(100);

MaxSingle(4,0);

MaxSingle(3,255); //+ + + + + + + +

delay(100);

MaxSingle(3,0);

MaxSingle(2,255); //+ + + + + + + +

delay(100);

MaxSingle(2,0);

MaxSingle(1,255); //+ + + + + + + +

delay(100);

MaxSingle(1,0);

MaxSingle(0x0c,0x00);

delay(300);

MaxSingle(0x0c,0x01);

}

void setup()

{

Serial.begin(9600); //tốc độ dữ liệu là 9600 bps

pinMode(pinCLK,OUTPUT); //khai báo clock

pinMode(pinCS,OUTPUT); //khai báo load

pinMode(pinDIN,OUTPUT); //khai báo data

Init\_MAX7219(); //khởi tạo led

start=true;

}

void loop()

{

if(Serial.available() > 0) //bluetooth nhận dữ liệu

{

incomingByte = Serial.read(); //đọc dữ liệu bluetooth

if(incomingByte == '0')

Left();

else if(incomingByte == '1')

Right();

else if(incomingByte == '2')

Head();

else if(incomingByte == '3')

{

Head();

start=false;

}

}

else if(start)

Begin();

delay(500);

}

1. **Giao tiếp rung**

const int motorPin = 7; //khai báo pin 7

void setup()

{

Serial.begin(9600); //tốc độ dữ liệu là 9600 bps

pinMode(motorPin, OUTPUT); //khai báo pin motor rung

}

void loop()

{

if(Serial.available() > 0) //bluetooth nhận dữ liệu

{

Serial.read(); //đọc dữ liệu bluetooth

digitalWrite(motorPin, HIGH); //motor rung hoạt động

delay(400); //chờ 400 mili giây

digitalWrite(motorPin, LOW); //motor rung ngừng hoạt động

Serial.flush(); //xóa dữ liệu cache

}

delay(500); //chờ 500 mili giây

}

1. **HC-06 bluetooth module**

HC-06 bluetooth module cho phép vi điều khiển kết nối với các thiết bị ngoại vi như: smartphone, laptop, usb bluetooth... thông qua giao tiếp serial gửi và nhận tín hiệu 2 chiều.

Cho phép sử dụng nguồn từ 3.5V đến 6V.

Tần số làm việc: 2.4GHz.

Mặc định, HC-06 sử dụng tốc độ dữ liệu là 9600 bps và mật khẩu là 1234.

Kết nối HC-06 với Arduino:

|  |  |  |
| --- | --- | --- |
| **HC-06** | **Arduino** | **Chú thích** |
| VCC | 5V | Cấp nguồn cho HC-06 hoạt động. |
| GND | GND | Nối dây đất đến Arduino. |
| TXD | RX | Nối output/TXD HC-06 đến input/RX Arduino. |
| RXD | TX | Nối input/RXD HC-06 đến output/TX Arduino. |

Cơ chế hoạt động:

* Cấp nguồn cho HC-06 bluetooth module hoạt động từ pin 5V của Arduino.
* Tín hiệu ra (TX) của Arduino sẽ đi đến tín hiệu vào (RXD) của HC-06.
* Tín hiệu ra (TXD) của HC-06 sẽ đi đến tín hiệu vào (RX) của Arduino.

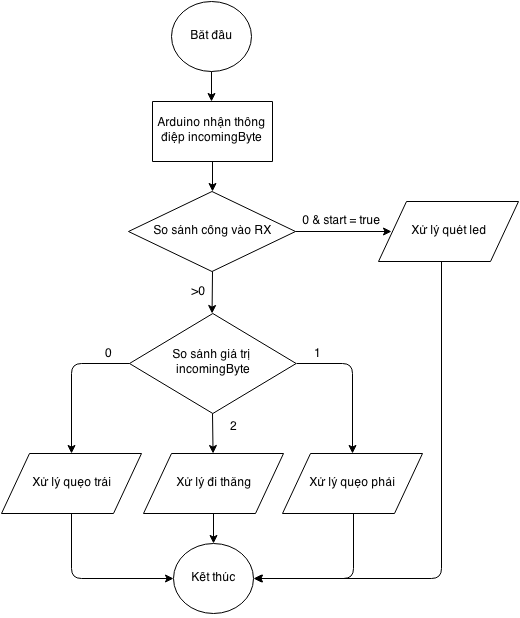
1. **Thông điệp trao đổi giữa điện thoại và Arduino**

Điện thoại sẽ tìm và ghép với HC-06 bluetooth.

Khi đã bắt tay được với nhau, Android sẽ gởi một thông điệp và chuyển đổi thông điệp này ra dạng byte để gởi đến HC-06.

HC-06 nhận thông điệp và đưa qua đường RX của Arduino để xử lý.

Arduino tiến hành chuyển đổi thông điệp vừa nhận ngược lại ra dạng ký tự để so sánh và đưa ra kết quả về tín hiệu đèn tương ứng.



Hình 2.7 - Sơ đồ khối led ma trận 8x8



Hình 2.8 - Sơ đồ khối giao tiếp rung

# CHƯƠNG III HƯỚNG PHÁT TRIỂN

Với những kết quả đạt được ở đồ án chuyên ngành, nhóm chúng em sẽ cố gắng huớng tới hiện thực hóa và đưa sản phẩm này đến với người tiêu dùng.

Cố gắng xây dựng các thiết bị điều hướng, tránh ùn tắc giao thông để phục vụ, giúp ích cho người dùng và cảnh sát trong những giờ cao điểm.

Lên kế hoạch kinh doanh sản phẩm ra thị trường trong thời gian sắp tới.

# CHƯƠNG IV TÀI LIỆU THAM KHẢO

[1] Reto Meier, 2012, Professional Android Application Development.

[2] Joe Pardue, 2005, C Programming for Microcontrollers, featuring ATMEL's AVR Butterfly and the free WinAVR Compiler, Smiley Micros.

[3] Ed Burnette, Hello Android, Introducing Google's Mobile Development Platform.

[4] Sams, Android Application Development in 24 Hours.