

---

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ

---



# **BÁO CÁO ĐỀ TÀI THIẾT KẾ HỆ THỐNG NHÚNG XE ĐIỀU KHIỂN TỪ XA BẰNG CỬ CHỈ TAY ÁP DUNG CẢM BIẾN GIA TỐC ADXL345**

*Thực hiện bởi:*

1. Võ Trọng Khôi - 1910284
2. Nguyễn Thành Hiệp - 1910186
3. Phan Minh Khang - 1911346

Giáo viên hướng dẫn: Ths Bùi Quốc Bảo

# MỤC LỤC

<b>CHƯƠNG 1: Ý TƯỞNG VỀ MÔ HÌNH SẢN PHẨM .....</b>	<b>2</b>
<b>1.1. Lí do chọn đề tài:.....</b>	<b>2</b>
<b>1.2. Use case: .....</b>	<b>2</b>
<b>1.3. Các yêu cầu: .....</b>	<b>3</b>
<b>CHƯƠNG 2: THIẾT KẾ HỆ THỐNG .....</b>	<b>3</b>
<b>2.1. Thiết kế nguyên lý:.....</b>	<b>3</b>
2.1.1. Sơ đồ khối:.....	3
2.1.2. Nguyên lý hoạt động tổng quát: .....	4
<b>2.2. Thiết kế kỹ thuật: .....</b>	<b>4</b>
2.2.1. Khối điều khiển: .....	4
2.2.2. Khối lái động cơ:.....	7
2.2.3. Khối hiển thị:.....	9
2.2.4. Cảm biến gia tốc ADXL345: .....	9
2.2.5. Module phát thu RF 433MHz: .....	11
2.2.6. Khối nguồn: .....	15
<b>2.3. Sơ đồ nguyên lý:.....</b>	<b>17</b>
<b>2.4. Sơ đồ mạch in: .....</b>	<b>20</b>
<b>2.5. Sơ đồ thuật toán: .....</b>	<b>21</b>
<b>2.6. Code và mạch mô phỏng proteus: .....</b>	<b>21</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>27</b>

## CHƯƠNG 1: Ý TƯỞNG VỀ MÔ HÌNH SẢN PHẨM

### 1.1. *Lí do chọn đề tài:*

Xe điều khiển từ xa là một đồ chơi giải trí phổ biến. Hiện nay có rất nhiều loại xe điều khiển với giá tiền dao động khoản mấy trăm ngàn Việt Nam đồng hoặc có thể hơn và được điều khiển thông qua các thiết bị điều khiển từ xa hay giao tiếp bằng giọng nói hoặc cử chỉ tay thông qua một thiết bị gắn trên bàn tay. Bên cạnh đó, người ta còn ứng dụng phát triển các sản phẩm xe điều khiển từ xa như trong việc giúp khám phá các hành tinh.

Nhóm chúng em muốn tạo một thiết bị giải trí giúp người sử dụng có thể thư giãn sau những giờ làm việc, học tập căng thẳng và thay vì sử dụng một thiết bị điều khiển từ xa là tay cầm có nút nhấn thì tại em sử dụng cử chỉ của bàn tay để điều khiển xe. Nhằm để nắm kiến thức môn học thiết kế hệ thống nhúng, nhóm muốn lựa chọn sản phẩm xe điều khiển với thiết bị điều khiển từ xa đeo trên bàn tay được điều khiển thông qua các cử chỉ tay để thuận tiện cho việc điều khiển và mới lạ hơn so việc điều khiển bằng giữ các nút trên hộp điều khiển.

### 1.2. *Use case:*

Di chuyển về các hướng theo cử động của bàn tay:

- Hướng tay xuống đất để tiến tới
- Hướng tay lên để lùi về sau
- Nghiêng tay qua trái để xe xoay sang trái tại chỗ
- Nghiêng tay sang phải để xe xoay sang phải tại chỗ
- Xe đứng yên khi mặt bàn tay úp xuống, song song với mặt đất

- Các trường hợp khác sẽ dừng lại không chạy

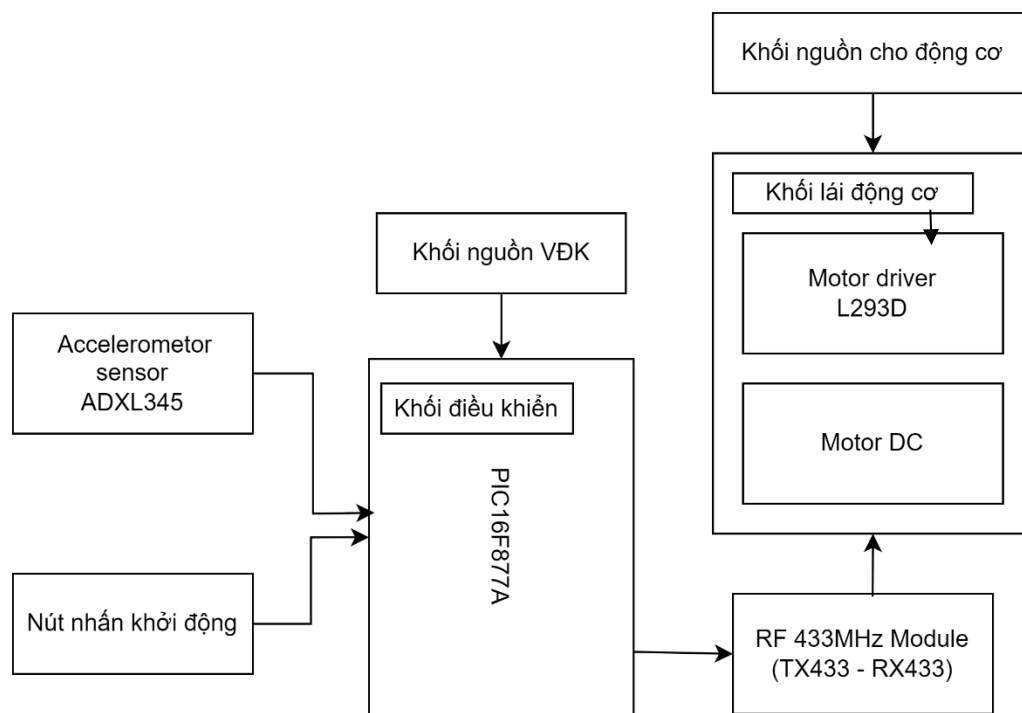
### 1.3. Các yêu cầu:

- Nhiệt độ thích hợp: -20 to +50 độ C
- Nguồn động cơ xe: dùng nguồn từ dưới 9V
- Nguồn thiết bị điều khiển: dùng nguồn từ dưới 9V
- Giá thành từ trên 300.000 VND
- Các rung động có thể chịu được: rung động khi xe đang chạy và thiết bị điều khiển rung lắc khi chuyển động tay.

## CHƯƠNG 2: THIẾT KẾ HỆ THỐNG

### 2.1. Thiết kế nguyên lý:

#### 2.1.1. Sơ đồ khối:



Khối điều khiển: nhận tín hiệu từ cảm biến gia tốc và đưa các tín hiệu điều khiển tới khối khác

Khối lái động cơ: nhận tín hiệu điều khiển từ vi xử lý thông qua module thu phát RF và điều khiển các động cơ DC

Khối hiển thị: dùng đèn LED hiển thị mức pin còn lại trên xe và trên bộ điều khiển

Khối nguồn: cung cấp nguồn nuôi vi xử lý, nguồn cho các IC và module

### *2.1.2. Nguyên lý hoạt động tổng quát:*

Cảm biến gia tốc được đặt trên tay người điều khiển, ghi lại cử chỉ của tay. Vi xử lý nhận tín hiệu từ cảm biến và tính toán tay đang ở vị trí như nào và đưa ra câu lệnh thích hợp. Câu lệnh này được đưa đến khối lái động cơ thông qua module phát thu RF. Các vị trí tương ứng với các lệnh tiến, lùi, sang trái, sang phải và đứng yên. Vì trên xe sử dụng nguồn là pin nên khối hiển thị sẽ dùng đèn LED đơn để hiển thị số pin còn lại của xe.

## **2.2. Thiết kế kỹ thuật:**

### *2.2.1. Khối điều khiển:*

Hiện tại, có nhiều họ vi điều khiển được sử dụng rộng rãi trên thị trường như:

Vi điều khiển 8051: có thiết kế đơn tinh thể Harvard, là dòng vi điều khiển có thiết kế cổ nhất được thiết kế đa dạng, và cải tiến cùng các công nghệ khác nhau như: MCS, CMOS. Dòng vi điều khiển này thường sử dụng ngôn ngữ lập trình C cho các chương trình ứng dụng thực tế. Đại diện của họ 8051 có AT89C51 là một vi điều khiển mạnh, công suất lớn, thích hợp dùng cho nhiều ứng dụng nhúng.

Vi điều khiển ARM: Dòng vi điều khiển sử dụng cấu trúc ARM với bộ nhớ 32 bits được sử dụng rộng rãi. Với những tính năng tiết kiệm năng lượng, sử dụng ngôn ngữ lập trình C. Hiện là vi điều khiển được sử dụng nhiều nhất trong các thiết bị di động, điều khiển từ xa.

Vi điều khiển AVR: Là dòng vi điều khiển do hãng Atmel sản xuất, dựa trên kiến trúc RISC. Với công nghệ cho phép thực hiện lệnh trong một kỳ xung nhịp. Vi điều khiển loại này có khả năng tối ưu hoá hiệu quả năng lượng sử dụng vẫn đảm bảo khả năng xử lý hiệu quả.

Vi điều khiển PIC: Là một họ của vi điều khiển RISC được sản xuất bởi Microchip Technology. Vi điều khiển PIC được thiết kế và cải tiến với nhiều ưu điểm vượt trội. Nhiều dòng PIC được sản xuất với modul tích hợp sẵn như USART, PWM, ADC,... Sử dụng ngôn ngữ lập trình và trình dịch khá đa dạng.

Khối điều khiển của sản phẩm này cần có các chức năng truyền thông I2C để giao tiếp với cảm biến gia tốc ADXL345, có chức năng truyền thông SPI để sử dụng cho module thu phát RF. Kết hợp với kiến thức lập trình vi điều khiển PIC được học trong môn “Thiết kế hệ thống nhúng” thì hệ thống này chọn sử dụng vi điều khiển PIC, cụ thể là PIC16F877A.

Sơ đồ chân PIC16F877A:

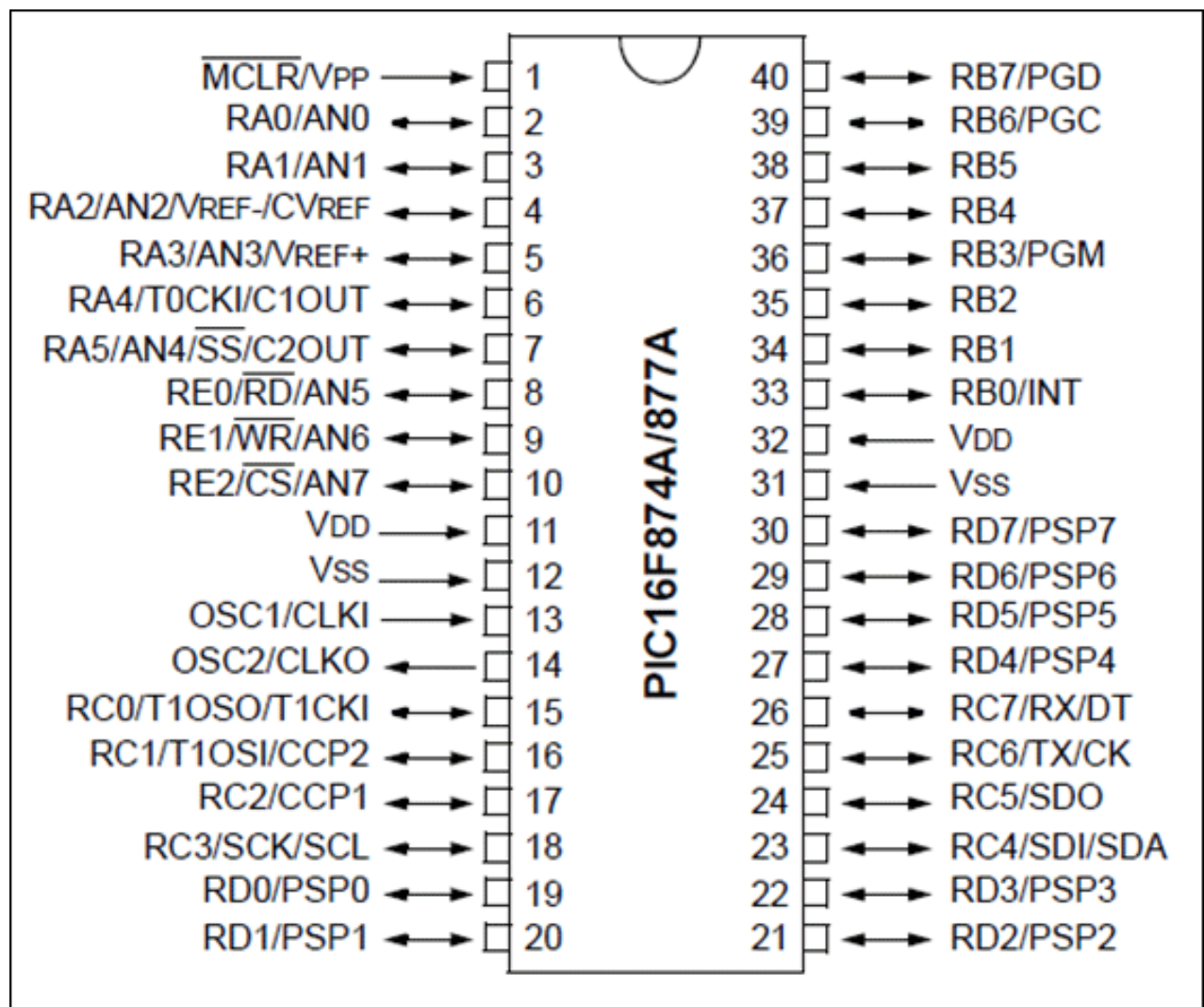


Figure 1. Sơ đồ chân PIC16F877A

Sử dụng bộ dao động tinh thể để tạo tần số hoạt động cho PIC 16F877A là 20MHz, gắn tại 2 chân OSC1 và OSC2, nối với tụ điện 22pF và nối đất.

Các chân Vdd được cấp vào nguồn nuôi 3.3V, các chân Vss được nối GND và có gắn thêm tụ bypass 100nF.

PIC16F877A có tổng cộng 5 Cổng là:

- Cổng A: có tổng cộng 6 chân bắt đầu từ chân số 2 đến chân số 7. Các chân cổng A được ký hiệu từ RA0 đến RA5 trong đó RA0 là ký hiệu của chân đầu tiên của Cổng A.

- Cổng B: có tổng cộng 8 chân bắt đầu từ chân số 33 đến chân số 40. Các chân cổng B được ký hiệu từ RB0 đến RB7 trong đó RB0 là ký hiệu của chân đầu tiên của cổng B.

- Cổng C: có tổng cộng 8 Chân. Các chân của nó không được thẳng hàng với nhau. Bốn chân đầu tiên của cổng C nằm ở chân số 15 đến chân số 18, còn bốn chân cuối cùng nằm ở chân số 23 đến chân số 26.

- Cổng D: có tổng cộng 8 chân. Các chân của nó cũng không thẳng hàng với nhau. Bốn chân đầu tiên của cổng D nằm ở chân số 19 đến chân số 22, trong khi bốn chân cuối cùng nằm ở chân số 27 đến chân số 30.

- Cổng E: có tổng cộng 3 chân bắt đầu từ chân số 8 đến chân số 10. Các chân cổng E được ký hiệu từ RE0 đến RE2 trong đó RE0 là ký hiệu của chân đầu tiên của cổng E.

Mỗi cổng của vi điều khiển PIC được liên kết với hai thanh ghi (register), ví dụ các thanh ghi cổng D là PortD và TRISD. Cả hai thanh ghi này đều có 8 bit vì cổng D có 8 chân. TRISD quyết định cổng là đầu ra hay đầu vào và chúng ta cũng có thể gán giá trị cho từng chân riêng biệt. Nếu chúng ta đã gán 0 thì nó sẽ là đầu ra và nếu chúng ta đã gán 1 thì nó sẽ là đầu vào. Ví dụ: nếu chúng ta gán  $TRISD = 0x01$ , thì 7 chân đầu tiên của cổng D sẽ là đầu ra nhưng chân cuối cùng sẽ là đầu vào vì 0x01 là 00000001 trong hệ nhị phân. Thanh ghi PortD chứa giá trị thực và giá trị này là sự kết hợp của tất cả 8 chân.

PIC16F877a có một cổng I2C có thể dễ dàng thực hiện giao tiếp I2C.

- Chân số 18 hoạt động như SCL, viết tắt của Serial Clock Line.
- Chân số 23 hoạt động như SDA, viết tắt của Serial Data Line.

PIC16F877a có một cổng nối tiếp trong đó được sử dụng để giao tiếp dữ liệu.

- Chân số 25 cũng hoạt động như TX vì vậy nếu bạn muốn thực hiện giao tiếp nối tiếp thì nó sẽ được sử dụng để gửi dữ liệu nối tiếp.
- Chân 26 cũng hoạt động như RX, vì vậy nếu bạn muốn thực hiện giao tiếp nối tiếp thì nó sẽ được sử dụng để nhận dữ liệu nối tiếp.

Trình biên dịch PIC16F877A được sử dụng là CCS C Compiler với các tính năng đơn giản, dễ sử dụng; dễ dàng chuyển đổi giữa tất cả các thiết bị MCU microchip PIC; giảm thiểu thời gian lập trình, dễ dàng xác định và thiết lập các ngắt, có các thư viện CCS miễn phí; có kiểu một bit tích hợp (Short Int) cho phép trình biên dịch tạo mã định hướng Bit rất hiệu quả.

### 2.2.2. Khởi lái động cơ:

Động cơ một chiều DC (DC là từ viết tắt của "Direct Current Motors") là Động cơ điều khiển bằng dòng có hướng xác định hay nói dễ hiểu hơn thì đây là loại động cơ chạy bằng nguồn điện áp DC- điện áp 1 chiều (Khác với điện áp AC xoay chiều). Đầu dây ra của động cơ thường gồm hai dây (dây nguồn- VCC và dây tiếp đất- GND). DC motor là một động cơ một chiều với cơ năng quay liên tục.

Các phương trình điều chỉnh tốc độ

- Thay đổi điện áp phản ứng
- Thay đổi điện trở mạch rotor
- Thay đổi từ thông

Trên thực tế phương pháp được sử dụng nhiều nhất là thay đổi điện áp phản ứng. Trong đó điển hình là phương pháp thay đổi độ rộng xung PWM. Đường đặc tính cơ của động cơ khi thay đổi điện áp phản ứng như sau.

Hầu hết các loại mạch điều khiển động cơ DC như cầu H hay fet+relay, L298... đều dùng để thay đổi điện áp phản ứng đặt vào động cơ. Và đảo chiều động cơ bằng cách thay đổi chiều điện áp đặt lên phản ứng.





*Figure 2. IC lái motor L293D*

L293D về cơ bản là một IC trình điều khiển hay bộ điều khiển động cơ. Nó có hai mạch cầu H tích hợp có thể điều khiển đồng thời hai động cơ DC theo cả chiều kim đồng hồ và ngược chiều kim đồng hồ. Nó hoạt động như một bộ khuếch đại dòng cao vì nó lấy tín hiệu dòng điện thấp ở đầu vào và cung cấp tín hiệu dòng điện cao hơn ở đầu ra để điều khiển các tải khác nhau, ví dụ động cơ bước và động cơ DC. Các tính năng của nó bao gồm phạm vi điện áp nguồn đầu vào lớn, tín hiệu đầu vào chống nhiễu cao dòng điện đầu ra lớn,... Các ứng dụng thực tế phổ biến của nó bao gồm trình điều khiển động cơ bước, trình điều khiển relay, trình điều khiển động cơ DC,...

Sơ đồ chân L293D:



*Figure 3. Sơ đồ chân L293D*

Khi hoạt động, tất cả chân GND đều nối đất, hai chân nguồn gồm Vss nối với +5V để cung cấp điện áp cho IC. Còn lại, chân Vs cung cấp điện áp cho động cơ chạy, trong khoảng từ 4,5V đến 36V tùy vào loại động cơ. Các chân Enable (Enable 1,2 và Enable 3,4) được sử dụng để bật các chân đầu vào cho động cơ 1 và động cơ 2 tương ứng. Vì trong hầu hết các trường hợp, chúng ta sẽ sử dụng cả hai động cơ, cả hai chân đều được giữ ở mức cao theo mặc định bằng cách kết nối với nguồn + 5V. Các chân đầu vào Input 1,2 được sử dụng để điều khiển động cơ 1 và chân đầu vào Input 3,4 được sử dụng để điều khiển động cơ 2. Các chân đầu vào được kết nối với bất kỳ mạch kỹ thuật số hoặc vi điều khiển nào để điều khiển tốc độ và hướng của động cơ theo như bảng sau:

Input 1 = HIGH	Output 1 = HIGH	Động cơ 1 quay theo chiều kim đồng hồ
Input 2 = LOW	Output 2 = LOW	
Input 3 = HIGH	Output 1 = HIGH	Động cơ 2 quay theo chiều kim đồng hồ
Input 4 = LOW	Output 2 = LOW	

Input 1 = LOW	Output 1 = LOW	Động cơ 1 quay ngược chiều kim đồng hồ
Input 2 = HIGH	Output 2 = HIGH	
Input 3 = LOW	Output 1 = LOW	Động cơ 2 quay ngược chiều kim đồng hồ
Input 4 = HIGH	Output 2 = HIGH	

Input 1 = HIGH	Output 1 = HIGH	Động cơ 1 đứng yên
Input 2 = HIGH	Output 2 = HIGH	
Input 3 = HIGH	Output 1 = LOW	Động cơ 2 đứng yên
Input 4 = HIGH	Output 2 = HIGH	

### 2.2.3. Khởi hiển thị:

Để hiển thị pin còn lại, hệ thống sử dụng đèn LED đơn. Khi pin còn lại dưới 30%, LED sẽ sáng.

### 2.2.4. Cảm biến gia tốc ADXL345:

Cảm biến gia tốc có thể được sử dụng để đo góc nghiêng cũng như xác định hướng của vật thể trong không gian. Tùy từng loại gia tốc kế, chúng ta có thể đọc dữ liệu theo 1 chiều (trục), 2 chiều hoặc 3 chiều. Trong bài này, gia tốc kế ADXL345 cung cấp thông tin 3 trục xyz. Cảm biến gia tốc kế thường biểu thị các phép đo bằng “g” hoặc trọng lực. Một “g” là giá trị của lực hấp dẫn trái đất bình phương bằng  $9,8 \text{ m/s}^2$ .



*Figure 4. Cảm biến gia tốc ADXL345*

Thông số kỹ thuật:

- Nguồn sử dụng: 3~5VDC
- Dòng sử dụng: 30uA
- Điện áp giao tiếp: 3~5VDC
- Chuẩn giao tiếp I2C / SPI
- Measuring Ranging:  $\pm 2g \pm 16g$
- 3-axis,  $\pm 2g / \pm 4g / \pm 8g / \pm 16g$
- Compact Accelerometer/Inclinometer
- Low Power Consumption

Nếu chúng ta có một gia tốc kế được đặt bằng phẳng, với trục z của nó hướng lên trên, ngược chiều với lực hấp dẫn, thì đầu ra trục Z của cảm biến sẽ là 1g. Mặt khác, đầu ra X và Y sẽ bằng 0, vì lực hấp dẫn vuông góc với các trục này và hoàn toàn không ảnh hưởng đến chúng. Nếu chúng ta lật ngược cảm biến, thì đầu ra trục Z sẽ là -1 g. Điều này có nghĩa là đầu ra của cảm biến do định hướng của nó đối với trọng lực có thể thay đổi từ -1g đến +1g.

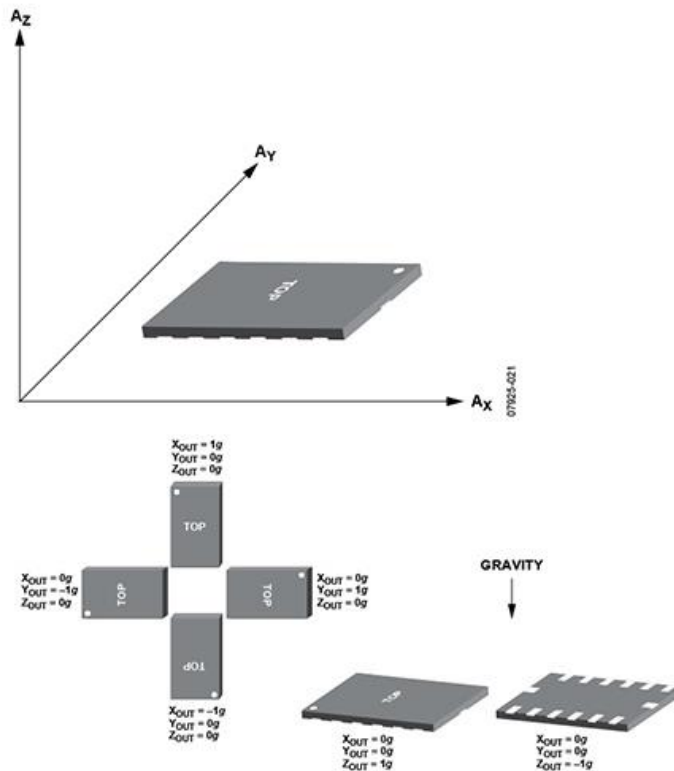


Figure 5. Hoạt động của cảm biến gia tốc

Cảm biến gia tốc này sử dụng giao tiếp I2C để giao tiếp với vi điều khiển, cho nên cần 2 dây để kết nối là SCL và SDA. Ngoài ra, còn cần thêm 2 dây cấp nguồn cho nó. Địa chỉ I2C của cảm biến ADXL345 có thể tìm được trong datasheet (0x53). Dữ liệu của mỗi trục được lưu trữ trong 2 byte đặt ở 2 thanh ghi. Các giá trị đầu ra phụ thuộc vào độ nhạy đã chọn, thay đổi từ  $\pm 2g$  đến  $\pm 16g$ . Độ nhạy mặc định là  $\pm 2g$ , đó là lý do tại sao chúng ta cần chia đầu ra cho 256 để nhận các giá trị từ -1 đến +1g. 256 LSB / g có nghĩa là chúng ta có 256 số đếm trên mỗi g.

Từ tính chất của hệ thống, việc cảm biến điều khiển yêu cầu độ nhạy  $\pm 2g$  nên có thể sử dụng cảm biến gia tốc ADXL345 cho hệ thống.

#### 2.2.5. Module phát thu RF 433MHz:

Hiện nay, trong việc truyền sóng vô tuyến qua một khoảng cách xa (vài trăm mét) ở các thiết bị dân dụng như remote ô tô, mở cửa cuốn, nhà thông minh, truyền dữ liệu giữa các vi điều khiển, .... RF 433Mhz là tín hiệu sóng vô tuyến có tần số 433Mhz. RF 433Mhz (hay 315Mhz) nằm trong miền tần số sóng điện từ UHF nên thường dùng để truyền tín hiệu trong môi trường không khí. Loại sóng này cũng tuân theo các định

luật phản xạ, khúc xạ, giao thoa của sóng điện từ và còn có khả năng đâm xuyên (xuyên tường, xuyên vật cản không phải là kim loại). Cụ ly truyền phụ thuộc vào nhiều yếu tố: như tần số truyền (tần số càng thấp truyền càng xa); độ ẩm không khí hoặc tính đồng nhất của môi trường truyền; phân cực sóng; công suất phát (dBm), độ nhạy máy thu (dBm)...

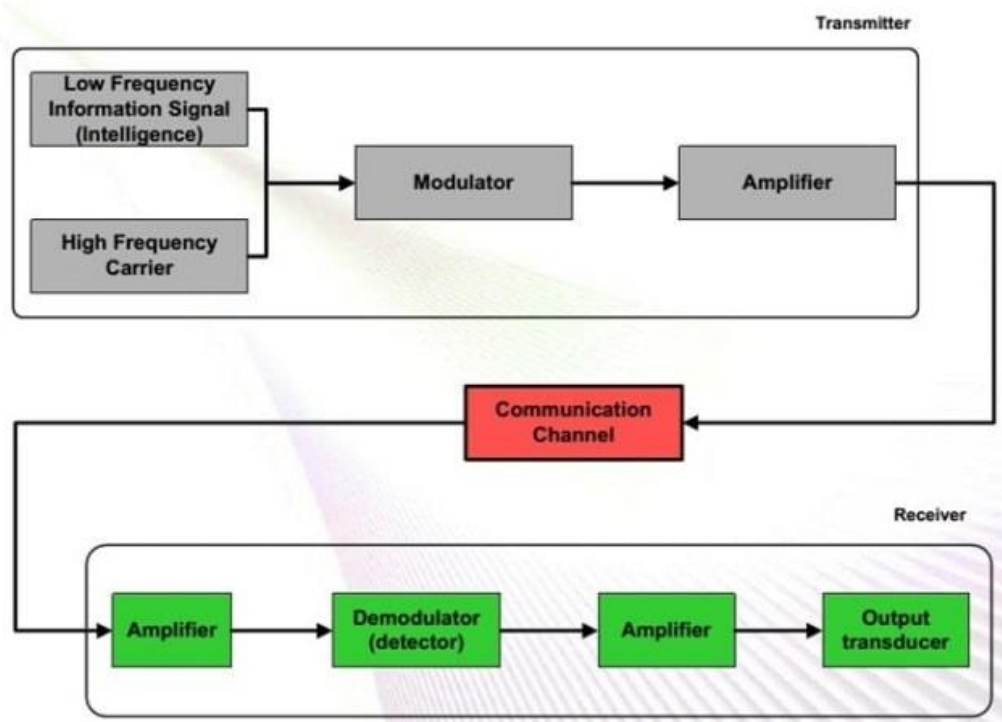


Figure 6. Sơ đồ mô hình phát thu sóng RF

Module thu phát RF 433Mhz là 1 module giá rẻ, dùng sóng rdio để truyền tín hiệu giữa bộ phát (transmitter) và bộ thu (receiver).

## BỘ THU

- Model: MX-05V
- Điện áp hoạt động: 5VDC
- Dòng hoạt động: 4mA
- Tần số thu: 433Mhz
- Kích thước: 30 \* 14 \* 7mm
- Anten cần lắp thêm: dài 32cm đơn lõi, có thể cuộn tròn.

## BỘ PHÁT

- Model: MX-FS-03V
  - Khoảng cách không có Anten: 20-30cm
  - Tùy thuộc vào Anten Và Môi Trường Truyền. Khoảng cách từ 50-100M. Có thể đạt tới khoảng cách 500M (điện áp cấp càng cao, truyền càng xa)
- Điện áp cấp: 3.5-12VDC
- Kích thước: 19\*19mm
- Tốc độ truyền: 4Kb/s
- Công suất truyền: 10mW
- Tần số truyền: 433Mhz
- Anten cần gắn thêm: dài 25cm đơn lõi có thể cuộn tròn.
- Thứ tự chân: DATA, VCC, GND.



*Figure 7. Module phát thu RF 433Mhz*

Trước khi được truyền từ vi điều khiển đến module phát, thông tin sẽ được đi qua một mạch encode để chuyển từ dữ liệu song song sang nối tiếp. IC được sử dụng để thực hiện công việc này là HT12E.

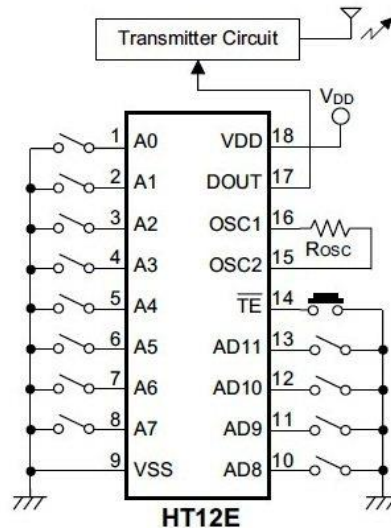


Figure 8. Sơ đồ chân HT12E

Thông số kỹ thuật HT12E:

- Hiệu ứng nhãn: Holtek
- Kiểu lắp: Thông qua lỗ
- Loại gói: PDIP
- Số lượng pin: 18
- Kích thước: 23.37 x 7.11 x 4.95 mm
- Điện áp cung cấp hoạt động tối đa: 12 V
- Nhiệt độ hoạt động tối đa: +75 ° C
- Nhiệt độ hoạt động tối thiểu: -20 ° C
- Điện áp cung cấp hoạt động tối thiểu: 2.4 V

HT12E thường được sử dụng trong các mạch RF hoặc IR, có tác dụng mã hóa 12 bit song song thành nối tiếp. 12 bit này chia thành 4 bit dữ liệu và 8 bit địa chỉ. Ở hệ thống bài này, ta xem như 8 bits địa chỉ là 0x00h, tức là nối đất 8 chân địa chỉ. Chân TE được xem là chân cho phép và ở mức tích cực thấp. Khi có tín hiệu cho phép ở mức tích cực, chân Dout bắt đầu một chu kỳ truyền 4 words và lặp lại liên tục nếu vẫn có tín hiệu cho phép tích cực.

Mặt khác, ở bên phần thu tín hiệu của module cũng sẽ cần một IC dùng để giải mã dữ liệu nối tiếp thành song song lại, là IC HT12D.

Thông số kỹ thuật:

- IC giải mã 12-bit dùng chung với HT12E
- Dữ liệu giải mã được gồm 8 bit địa chỉ và 4 bit dữ liệu (8+4=12-bits)
- Dùng trong truyền thông không dây RF hoặc IR
- Điện áp hoạt động là 5V
- Dòng điện stand by thấp, 0.1uA tại  $V_{cc}=5V$
- Có 2 loại 16 chân DIP và 20 chân SOP

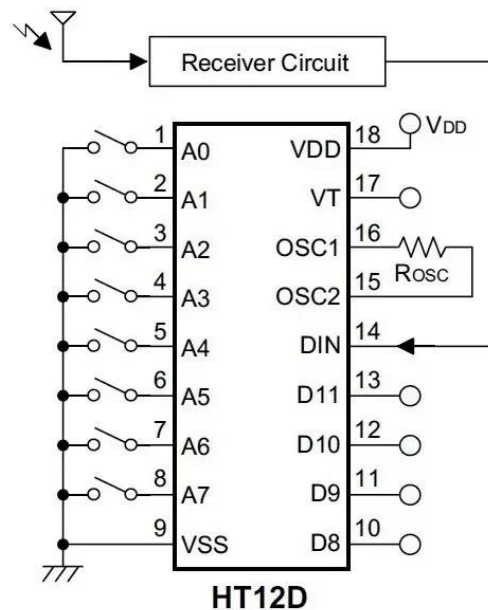


Figure 9. Sơ đồ chân HT12D

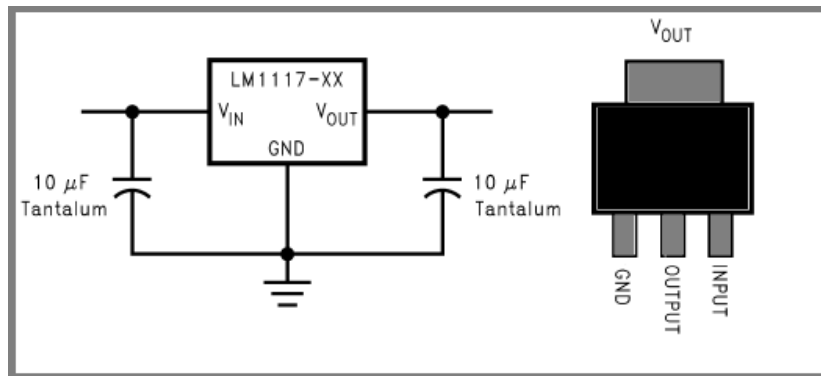
Để hoạt động IC HT12D cần nguồn nuôi thông qua chân Vdd. Chân  $V_T$  được giữ ở mức cao để xác định truyền hợp lệ. Cả 2 IC HT12E và HT12D đều có một bộ tạo dao động nội và có thể dùng một điện trở nối giữa 2 chân OSC1 và OSC2 là có thể hoạt động được. Các chân địa chỉ được nối đất để có địa chỉ là 0x00h, giống với địa chỉ từ HT12E đã định trước đó.

#### 2.2.6. Khối nguồn:

Hệ thống này yêu cầu 2 khối nguồn độc lập bao gồm một nguồn dành cho phần điều khiển và một nguồn dành cho phần động cơ. Nguồn cấp chính là từ pin, thông qua các mạch hạ áp để đưa vào vi điều khiển và IC. Cảm biến gia tốc ADXL345 cần một



điện áp tham chiếu cho giao thức I2C (digital interface supply voltage), nên hệ thống sẽ cần sử dụng một mạch hạ áp 3.3V dùng IC LM1117.



*Figure 10. Mạch hạ áp 3.3V*

Thông số kỹ thuật IC LM1117

- Bộ ổn áp tuyến tính 3 cực cố định hoặc có thể điều chỉnh
- Loại điện áp cố định: 1.8V, 2.5V, 3.3V và 5V
- Phạm vi điện áp thay đổi: 1,25V đến 13,8V
- Dòng điện đầu ra là 800mA
- Tích hợp giới hạn dòng điện và bảo vệ nhiệt.
- Nhiệt độ lớp tiếp giáp hoạt động là 125 ° C
- Gói TO-220, SOT223, TO263

Ngoài ra, vi điều khiển PIC16F877A sử dụng nguồn nuôi 5V nên hệ thống cần thêm một mạch hạ áp 5V sử dụng IC LM7805.

Thông số kỹ thuật:

- Điện áp vào: DC/AC 12V-24V (Không phân biệt âm dương)
- Điện áp ra: 5 VDC
- Dòng ra tối đa: 1.2A

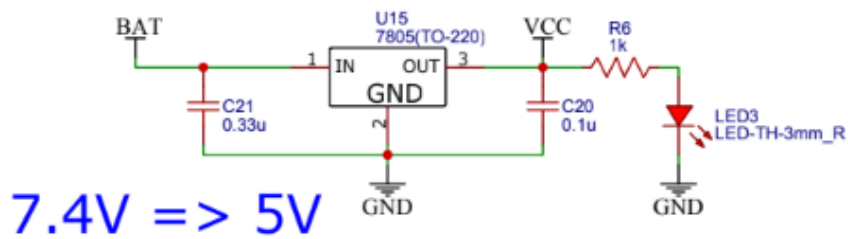


Figure 11. Mạch hạ áp 5V

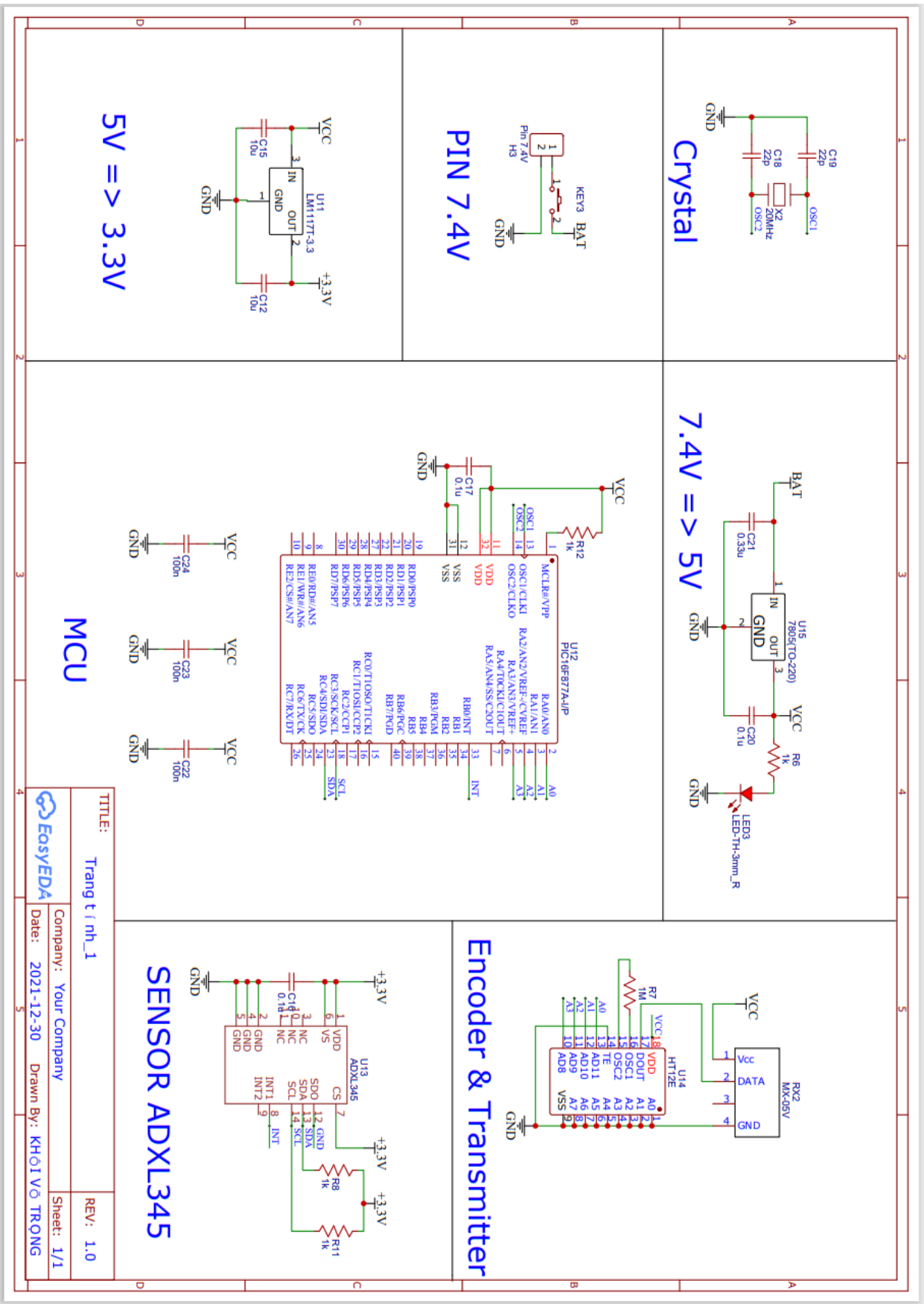
Các tụ lọc có chức năng khử nhiễu cho đầu ra 5V của mạch hạ áp.

Bảng giá các linh kiện sử dụng:

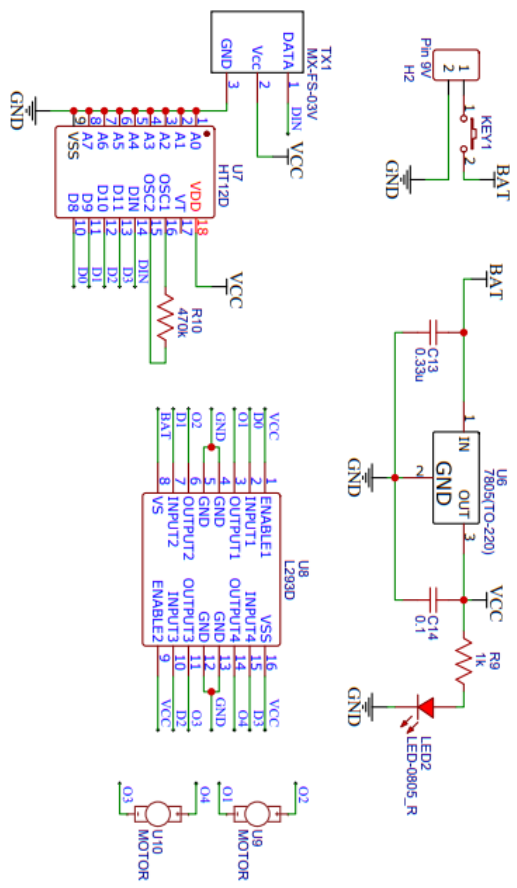
Tên linh kiện	Số lượng	Giá tiền
PIC16F877A	1	42.000đ
Module phát thu RF 433MHz	1	20.000đ
HT12E. HT12D	2	17.000đ
Module lái động cơ L293D	1	20.000đ
Cảm biến ADXL345	1	45.000đ
LM1117	1	3.000đ
LM7805	1	3.000đ
Pin Lithium 3.7V	2	40.000đ
Pin 9V	1	14.000đ
Động cơ DC	2	26.000đ
Điện trở các loại	10	3.000đ
Tụ điện các loại	15	3.000đ
Thạch anh 20MHz	1	4.000đ

### 2.3. Sơ đồ nguyên lý:

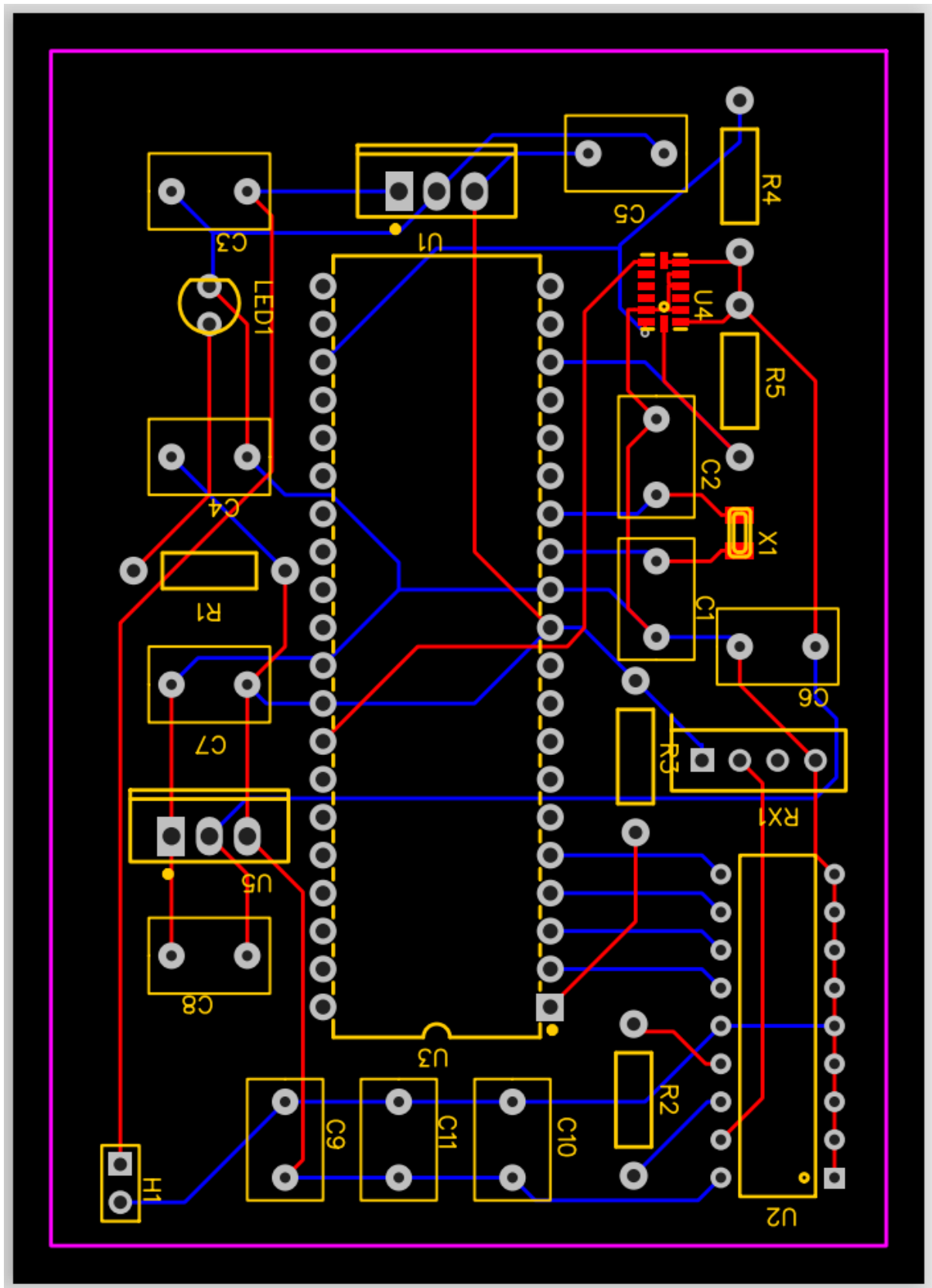
Mạch phát tín hiệu điều khiển xe:



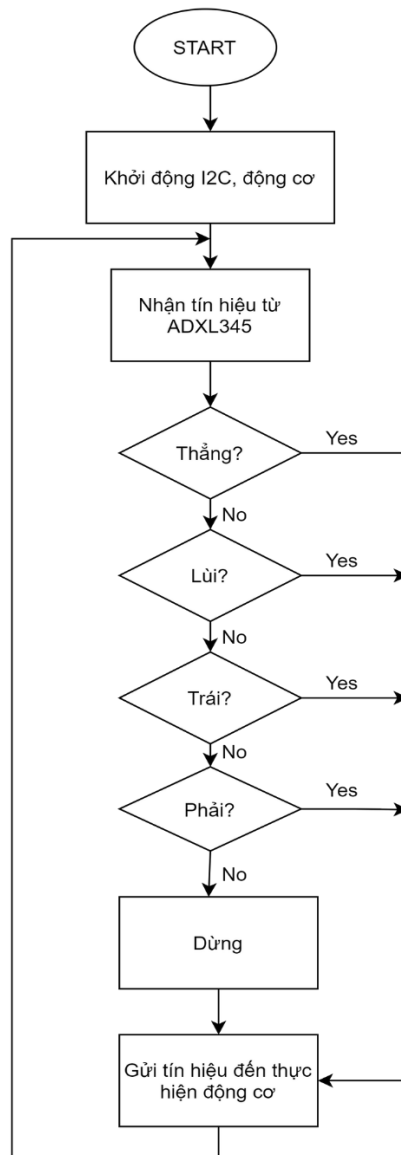
Mạch thu tín hiệu và lái động cơ:



2.4. Sơ đồ mạch in:



## 2.5. Sơ đồ thuật toán:



## 2.6. Code và mạch mô phỏng proteus:

Code để nạp vào mạch như sau:

```
#include <16F877A.h>
#use delay(clock = 20000000)
#use i2c(master, sda = PIN_C4, scl = PIN_C3)
void write(int diaChi_i2c, byte address, byte val);
byte read(int diaChi_i2c, byte address);
void send_sig(int x, int y);
```

```

void init_adxl345()//cau hinh adxl345
{
    write(0x53,0x38,0x9f);//che do Stream, 32 samples
    write(0x53,0x31,0x00);//10bit, range +-2g
    write(0x53,0x2C,0x0d);//BW400, f_output 800
    delay_us(10);
    write(0x53,0x2D,0x08);//bat dau do
}

void main()
{
    init_adxl345();
    while(true) {
        int16 x,y;
        //lay gia tri x,y
        x=(read(0x53,0x33)*256) | (read(0x53,0x32));
        x = x*2/1024;
        y=(read(0x53,0x35)*256) | (read(0x53,0x34));
        y = y*2/1024;
        send_sig(x, y);
    }
}

//ham viet gia tri vao
void write(int diaChi_i2C, byte address, byte val) {
    i2c_start();
    i2c_write(diaChi_i2C<<1);
    i2c_write(address);
    i2c_write(val);
    i2c_stop();
}

//ham doc gia tri ra

```

```

byte read(int diaChi_i2c, byte address) {
    int value;
    i2c_start();
    i2c_write(diaChi_i2c<<1);
    i2c_write(address);
    i2c_start();
    i2c_write(diaChi_i2c<<1|0x01);
    value=i2c_read(0);
    i2c_stop();
    return value;
}

void send_sig(int x, int y){
    if (y > 0.3 ){ //forward
        output_B(0x05);
        delay_ms(10);
    }
    else if (y < -0.3){ //backward
        output_B(0x0A);
        delay_ms(10);
    }
    else if (x < -0.3){ //left
        output_B(0x0D);
        delay_ms(10);
    }
    else if (x > 0.3){ //right
        output_B(0x07);
        delay_ms(10);
    }
    else if (-0.3 < x < 0.3 && -0.3 < y < -0.3) { //stay
        output_B(0x0F);
    }
}

```



```
delay_ms(10);
```

```
}
```

```
}
```

Sử dụng phần mềm Proteus 8.13 để thực hiện mô phỏng phần truyền tín hiệu RF giữa mạch phát và mạch thu. Mạch thực hiện được mắc như hình dưới.

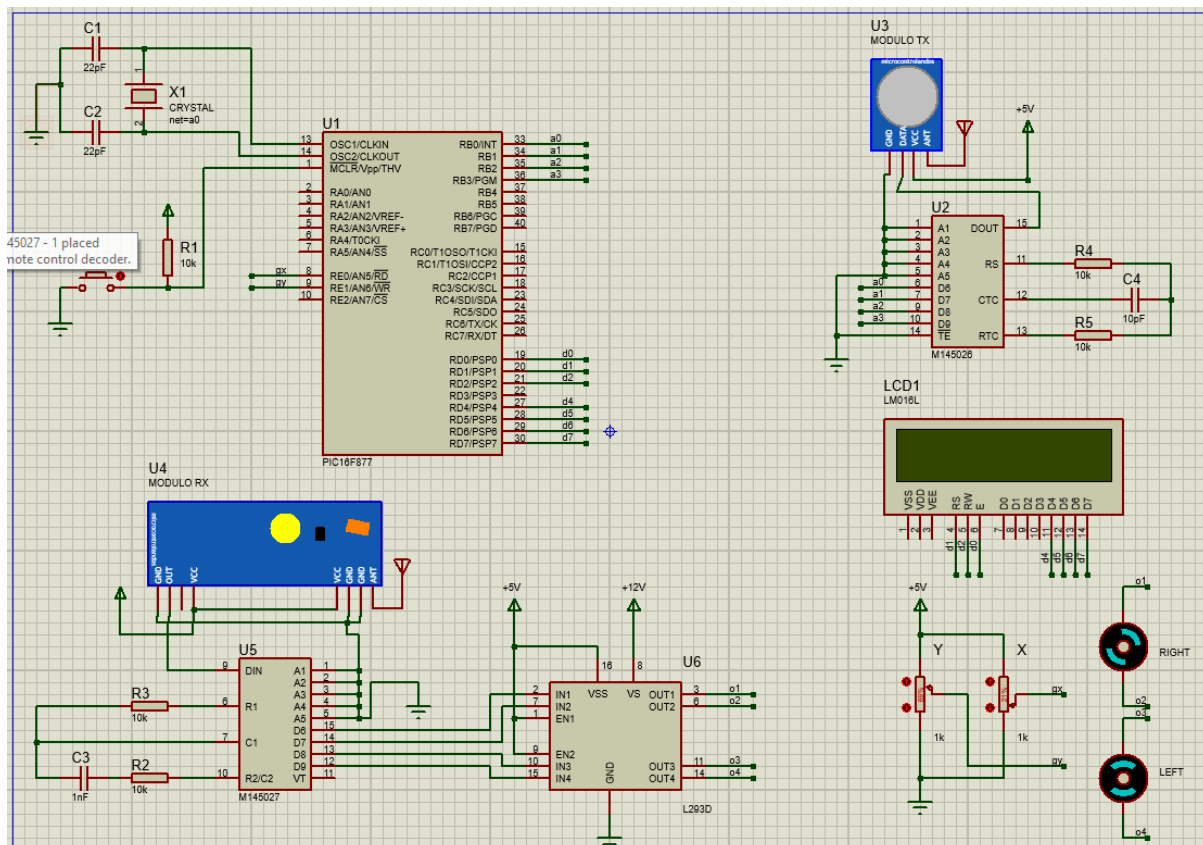


Figure 12. Mạch mô phỏng proteus

Code mô phỏng trên proteus sẽ khác với code nạp vào mạch thực tế. Do không mô phỏng được hoạt động của cảm biến gia tốc nên cảm biến này được thay bằng biến trở đưa vào các chân analog của PIC. Giá trị của biến trở sẽ được giả sử tương ứng với giá trị của cảm biến gia tốc là từ -1g đến 1g. Từ đó, dùng giá trị giả sử đó để tạo ra lệnh điều khiển gửi tới module lái động cơ.

Code:

```
#include <16f877.h>
#define adc = 10
#define delay(clock = 20000000)
#include <lcd.c>
void send_sig(float x, float y);
void main()
{
    lcd_init();
    lcd_gotoxy(1,1);
    printf(lcd_putc, "gx: ");
    lcd_gotoxy(1,2);
    printf(lcd_putc, "gy: ");
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(ALL_ANALOG);
    while(1)
    {
        set_adc_channel(5); //chon doc ADC chan AN5
        delay_ms(10);
        float gx = read_adc()*2.0/1024.0 - 1;
        delay_us(10);
        set_adc_channel(6); //chon doc ADC chan AN6
        delay_ms(10);
        float gy = read_adc()*2.0/1024.0 - 1;
        delay_us(10);
        lcd_gotoxy(5,1);
        printf(lcd_putc, "%f", gx);
        lcd_gotoxy(5,2);
        printf(lcd_putc, "%f", gy);
        send_sig(gx, gy);
    }
}
```

```

    }
}
void send_sig(float x, float y)
{
    if (y > 0.3 ){ //forward
        output_B(0x05);
        delay_ms(10);
    }
    else if (y < -0.3){ //backward
        output_B(0x0A);
        delay_ms(10);
    }
    else if (x < -0.3){ //left
        output_B(0x0D);
        delay_ms(10);
    }
    else if (x > 0.3){ //right
        output_B(0x07);
        delay_ms(10);
    }
    else { //stay
        output_B(0x0F);
        delay_ms(10);
    }
}

```

## TÀI LIỆU THAM KHẢO

Điện tử tương lai, “GIỚI THIỆU VỀ PIC 16F877A”, <https://dientutuonglai.com/gioi-thieu-pic-16f877a.html>, Ngày truy cập 29/12/2021.

Điện tử tương lai, “Tìm hiểu IC LM1117”, <https://dientutuonglai.com/tim-hieu-lm1117.html>, Ngày truy cập 29/12/2021.

hiepl902 (2017), “Các loại vi điều khiển nhúng hiện nay”, <https://www.standrivel.com/cac-loai-vi-dieu-khien-nhung-hien-nay/>, Ngày truy cập 29/12/2021.

Lazylynx (2021), “B38 – Sử dụng gia tốc kế điện tử ADXL345”, <https://duannho.com/2021/04/03/b38-su-dung-gia-toc-ke-dien-tu-adxl345/>, Ngày truy cập 29/12/2021.

Nguyễn Hữu Phước (2019), “Điều khiển động cơ DC dùng IC L293D và Arduino”, <https://dientuviet.com/dieu-khien-dong-co-dc-l293d-va-arduino/>, Ngày truy cập 29/12/2021.

Nguyễn Hữu Phước (2019), “Giới thiệu vi điều khiển PIC 16F877A”, <https://dientuviet.com/gioi-thieu-pic-16f877a/>, Ngày truy cập 29/12/2021.

Rajkumar Sharma (2015), “3.3V VOLTAGE REGULATOR”, <https://www.electronics-lab.com/project/3-3v-voltage-regulator/>, Ngày truy cập 29/12/2021.

Randy (2018), “Kỹ thuật sóng RF 433Mhz và 315Mhz”, <https://quyluan.com/ky-thuat-song-rf-433mhz-va-315mhz/>, Ngày truy cập 29/12/2021.