**MỤC LỤC**

**Chương 1.** **MỞ ĐẦU 1**

**Chương 2.** **CÔNG NGHỆ SRF-05 VÀ CƠ SỞ LÝ THUYẾT 2**

2.1 SRF-05 là gì? 2

2.2 Lịch sử hình thành của SRF-05. 2

2.3 Các thành phần của một hệ thống SRF-05. 3

2.4 Nguyên lý hoạt động của SRF-05. 6

2.5 Các dải tần số hoạt động của hệ thống SRF-05. 6

2.6 Các phần mềm sử dụng 9

**Chương 3.** **THIẾT KẾ HỆ THỐNG 10**

3.1 Các khối chính và chức năng 10

3.2 Mô hình kết nối 11

3.3 Mạch phát triển Arduino Uno Nano 11

*3.3.1* *Giới thiệu chung về Arduino 11*

*3.3.2* *Lập trình cho Arduino 16*

3.4 Màn Hình Nhân Dữ Liệu LCD 17

*3.4.1* *Các tính năng 17*

3.5 Các Chi Tiết Kỹ Thuật 19

*3.5.1* *Cấu tạo 20*

*3.5.2* *Tiêu chuẩn kĩ thuật 20*

*3.5.3* *Sơ đồ khối 21*

*3.5.4* *Tổ chức dữ liệu 22*

3.6 Kết nối phần cứng 22

**Chương 4.** **Hệ thống điều khiển 24**

4.1 Lưu đồ thuật toán trên bộ xử lý 24

4.2 Lưu đồ hệ thống phần mềm 25

4.3 Code chương trình nạp mạch Arduino 25

4.4 Giao diện hệ thống 28

**Chương 5.** **KẾT LUẬN 29**

**TÀI LIỆU THAM KHẢO 29**

* **MỞ ĐẦU**

Ngày nay, công nghệ thông tin đã phát triển với tốc độ nhanh chóng, nó đã trở thành một ngành công nghiệp mũi nhọn trên thế giới cũng như ở Việt Nam. Việc tin học hoá đã được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học công nghệ, kinh tế, quân sự và nhiều ngành, nghề khác.

Trên thị trường hiện nay, các doanh nghiệp đang phải tăng cường tối đa hiệu quả cung cấp các dịch vụ của mình, để đạt được mục tiêu này, họ phải dựa vào một trong những tài sản lớn nhất của mình đó là nguồn nhân lực. Các công tác quản lý, chấm công theo phương pháp thủ công đã không còn khả năng đáp ứng được nhu cầu thực tế công việc đòi hỏi, do đó việc ứng dụng tin học vào trong công tác quản lý là hết sức cần thiết, nó đạt hiệu quả cao về thời gian, không gian đem lại lợi ích kinh tế, sự thuận tiện cho người sử dụng và đảm bảo tính chính xác, minh bạch về thông tin.

Mặt khác, công nghệ “Nhận dạng đối tượng bằng số Sonar” đang trở lên phổ biến, các thiết bị giá thành ngày càng rẻ. Với ưu điểm đó em đã lựa chọn đề tài: “ Ứng dụng công nghệ đo khoảng cách an toàn từ đó giúp cho các người khiếm thị và các máy móc như ô tô hay một số máy móc cao cấp hơn như tàu ngầm đều sử dụng để tiếp thu và xử lý các vật thể trước mặt.

Mục tiêu của đề tài nhằm xây dựng hệ thống bao gồm các khối phần cứng và phần mềm đi kèm đáp ứng về nhu cầu và một số các trường hợp đo đặc biệt ở những địa hình nguy hiểm.

Nội dung của báo cáo bao gồm 5 chương. Chương I giới thiệu đề tài và mục đích đề tài; chương II tình bày về công nghệ Đo khoảng cách bằng sống Sonar và các cơ sở lý thuyết có liên quan; chương III trình bày về cấu trúc, chức năng của các thành phần cũng như chi tiết các thiết bị sử dụng; chương IV mô tả hệ thống điều khiển cũng như phần mềm hệ thống; chương V trình bày kết luận và hướng phát triển tiếp theo của đề tài.

* **CÔNG NGHỆ SRF-05 VÀ CƠ SỞ LÝ THUYẾT**
* **SRF-05 là gì?**
* Cảm biến siêu âm HY-SRF05 là cảm biến dùng để đo khoảng cách tới vật cản nhờ vào sóng siêu âm được phát ra từ cảm biến và nhận về. Khoảng cách có thể đo trong khoảng 2-300 cm với độ chính xác gần như chỉ phụ thuộc vào cách lập trình,Công nghệ Sóng tàu ngầm cũng sử dụng nguyên lý này. Với độ bảo mật và đáng tin cậy, SRF-05 đã và đang được ứng dụng rất nhiều vào trong cuộc sống có thể thể kể đến như:

- Máy cho người khiếm thị

- Máy nhận biết khoảng cách an toàn cho ô tô

- Máy đo đặc biệt dành cho các công việc khác nhau như kỹ thuật xây dựng

Và rất nhiều ứng dụng trong nhiều lĩnh vực khác nhau…

* **Lịch sử hình thành của SRF-05.**

Tận Dụng Nguồn lợi từ sóng Sonar từ đó SRF-05 được trãi qua các giai đoạn phát triển như.

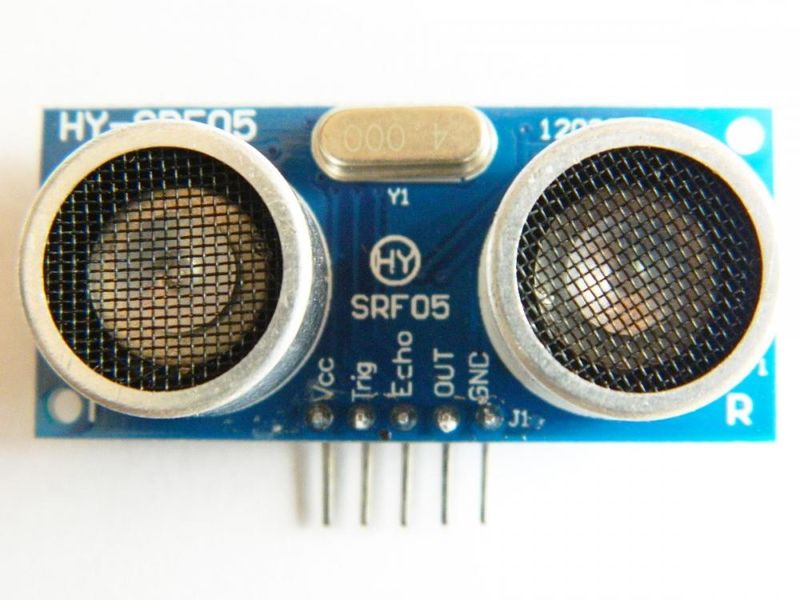
Ngày 23/01/1940, một thiết bị của Mario Cardullo được tạo ra. Nó là hình thức sơ khai đầu tiên của một thiết bị SRF-05 vì nó là một bộ tiếp sóng âm thanh thụ động với bộ nhớ. Các bằng sáng chế cơ bản Cardullo bao gồm việc sử dụng SN, âm thanh và ánh sáng như là phương tiện truyền dẫn. Các kế hoạch trình bày ban đầu để các nhà đầu tư trong năm 1969 cho thấy sử dụng công nghệ này có thể sử dụng trong giao thông vận tải

Một cuộc thí nghiệm đầu của các thẻ SRF-05, cả thụ động và bán thụ động, được thực hiện bởi Steven Depp, Alfred Koelle, và Robert Frayman tại Phòng thí nghiệm Quốc gia Los Alamos vào năm 1973. Các hệ sử dụng tần số 915 MHz và sử dụng thẻ 12-bit. Kỹ thuật này được sử dụng bởi phần lớn các SRF và máy thu sóng tags ngày nay.

* **Các thành phần của một hệ thống SRF-05.**

Dạng đơn thuần của một hệ thống SRF-05 được tạo nên bởi hai thành phần chính đó là thiết bị đọc - ghi (Reader/Writer) và máy thu sóng .

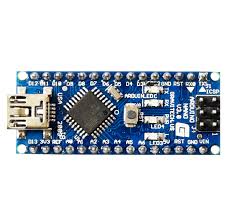
Thiết bị đọc - ghi (Reader): là một thiết bị giao tiếp không dây, có thể phát hiện các thẻ Tần số có cùng tần số hoạt động ở trong một phạm vi nhất định. Chức năng của thiết bị đọc - ghi là đọc, ghi dữ liệu được lưu trong Máy.



**Thiết bịCảm biến siêu âm HY-SRF05**



**Thiết bịCảm biến siêu âm CR-SRF05**



*ARDUINO NANO*

* **Sonar** (viết tắt từ [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *sound navigation and ranging*) là một kỹ thuật sử dụng sự lan truyền [âm thanh](https://vi.wikipedia.org/wiki/%C3%82m_thanh) (thường là dưới nước) để tìm đường di chuyển (tức *đạo hàng*), liên lạc hoặc phát hiện các đối tượng khác ở trên mặt, trong lòng nước hoặc dưới đáy nước, như các cá, tàu bè, vật thể trôi nổi hoặc chìm trong bùn cát đáy, v.v.

Có hai loại sonar:

Sonar chủ động thì tự phát xung sóng và nghe tiếng vọng lại.

Sonar bị động chỉ nghe âm thanh do tàu bè hay nguồn khác phát ra.

Sonar chủ động dùng đầu phát (Transmitter) phát xung sóng, thường được gọi là một

"ping", và nghe tiếng vọng lại ở đầu thu (Receiver). Có nhiều cách bố trí hình học các

đầu phát và thu, cho ra cách thức định vị đối tượng khác nhau:

Nếu phát và thu ở cùng một chỗ, hoạt động của nó là đơn tĩnh (monostatic) Nếu phát

và thu tách biệt, hoạt động của nó là song tĩnh (bistatic), Nếu có nhiều đầu phát (hoặc

nhiều đầu thu) ở vị trí tách biệt, hoạt động của nó là đa tĩnh (multistatic).

*Sonar thụ động* lắng nghe mà không phát tín hiệu. Nó thường được sử dụng trong quân

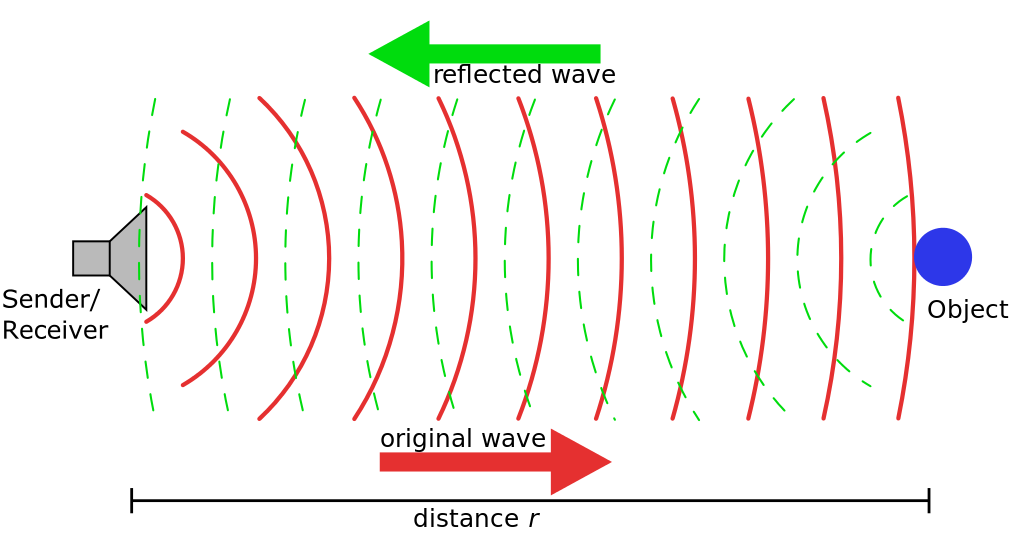
sự, mặc dù nó cũng được sử dụng trong các ứng dụng khoa học, ví dụ để phát hiện cá

trong nghiên cứu biển hoặc đánh cá, các vụ nổ mìn ở đảo, để nghe các vụ rung chấn hay

phun trào đáy biển. Theo nghĩa rộng Sonar thụ động bao gồm các kỹ thuật phân tích liên

quan đến âm thanh phát ra từ xa nhằm tới xác định vị trí và bản chất nguồn phát.

* **Nguyên lý hoạt động của SRF-05.**



*Nguyên lý hoạt động của một hệ thống SRF-05*

Thiết bị đọc phát ra sóng điện từ ở tần số nhất định qua ăng ten (đa số các thiết bị hiện nay đều có ăng ten tích hợp sẵn ngay trên thiết bị, chúng ta không cần lắp thêm), khi máy thu nằm trong vùng phát sóng của thiết bị đọc, nó sẽ nhận được năng lượng và phát lại từ chính vật thể trước mặt. Từ đó thiết bị đọc biết chính xác được thiết bị nào đang nằm trong vùng kiểm soát.

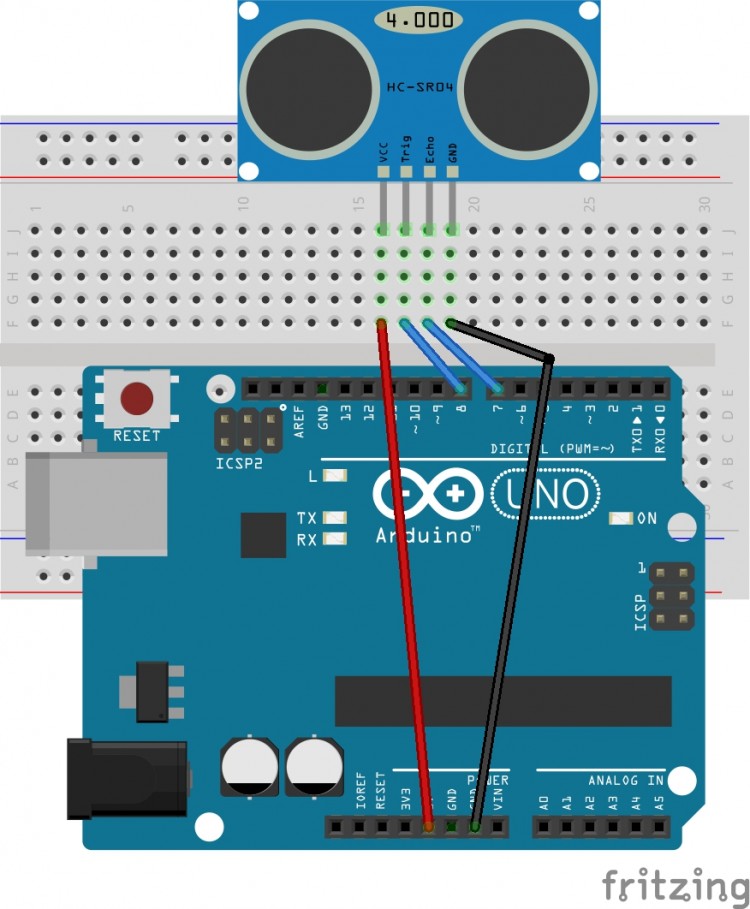
* **Các dải tần số hoạt động của hệ thống SRF-05.**

|  |  |  |  |
| --- | --- | --- | --- |
| Dải tần số | Khoảng cách | Tốc độ | Chú thích |
| 120–150 kHz | 10 cm | Chậm | Nhận dạng động vật, khoá phương tiện, các ứng dụng trong công nghiệp |
| 13.56 MHz | Tối đa 1 m | Trung bình | Thẻ thông minh  (ISO/IEC 15693, ISO/IEC 14443 A,B), kiểm soát đồ vật, điều khiển truy nhập |
| 433 MHz | Tối đa 2 m | Trung bình | Ứng dụng trong quốc phòng |
| 865 – 868 MHz  (Châu Âu)  902 – 928 MHz  (Bắc Mĩ) | Tối đa 3 m | Cao | Kiểm soát đồ vật |
| 2450 – 5800 MHz | Tối đa 100m | Cao | Ứng dụng trong các ngành công nghiệp |
| 3.1 – 10 GHz | 200 m | Cao | Ứng dụng trong các ngành công nghệ đặc biệt |

*Các tần số hoạt động của RFID*

*Truyền dữ liệu SPI*

* **Các phần mềm sử dụng**
* Phần mềm lập trình mạch phát triển Arduino: Arduino IDE
* Driver USB Arduino.
* **THIẾT KẾ HỆ THỐNG**
* **Các khối chính và chức năng**



*Các khối chính của hệ thống*

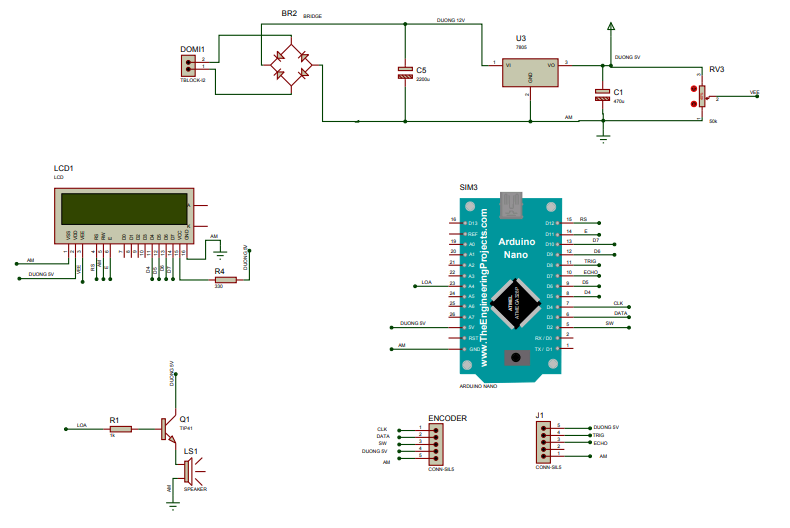
* Khối thiết bị thu : Thu Sóng hồi âm
* Khối vi xử lý: nhận dữ liệu từ thiết bị đọc, đóng gói và gửi chuỗi ID lên khối server, nhận lệnh từ server và thực thi những lệnh liên quan tới đầu ra.
* Khối server: tiếp nhận và xử lý thông tin từ khối vi xử lý gửi lên, thực hiện các truy vấn CSDL.

**Lựa chọn thiết bị:**

* Khối thiết bị đọc : sử dụng module SRF-05.
* Khối vi xử lý: sử dụng mạch phát triển Arduino Nano
* Khối server: sử dụng máy tính cá nhân.

**Ưu điểm của các thiết bị:**

* SRF-05: nhỏ gọn, dễ kết nối, không cần có ăng ten ngoài, tiện dụng vì thư viện được nhà sản xuất viết sẵn.
* Arduino Nano: nhỏ gọn, mạch được làm với hệ thống máy hiện đại, đảm bảo các mối hàn luôn được bảo vệ. Có sẵn khối chuyển đổi Serial – USB giao tiếp trực tiếp với máy tính không cần thêm bộ chuyển đổi RS232. Sử dụng trực tiếp nguồn điện từ cổng USB vì thế không cần lắp bổ sung thêm nguồn phụ. Dễ dàng tháo lắp, thay đổi chip xử lý (AVR họ 8 bit). Nạp chương trình trực tiếp thông qua chip ATmega16U2 gẵn trên mạch được nạp sẵn firmware. Ngôn ngữ lập trình dựa theo C/C++ đơn giản, dễ hiểu; chương trình lập trình tiện dụng, tích hợp sẵn Serial Monitor.
* **Mô hình kết nối**



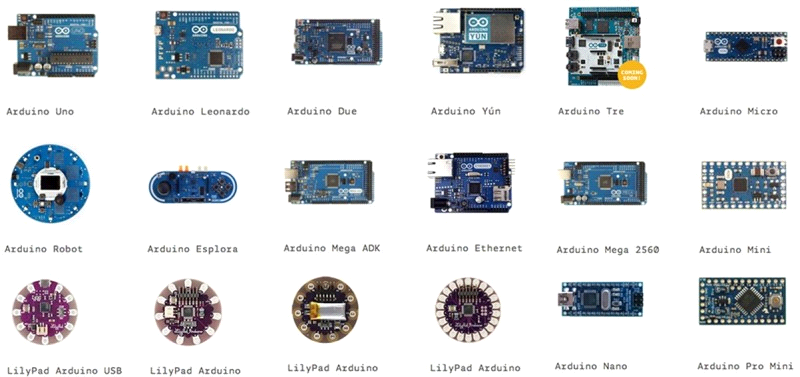
*Mô hình kết nối thực tế hệ thống*

* **Mạch phát triển Arduino Nano**
* **Giới thiệu chung về Arduino**

Mạch phát triển Arduino thế hệ đầu tiên được công bố năm 2006. Nó là công cụ cho sinh viên thực hành của giáo sư Massimo Banzi giảng dạy tại trường Interaction Design Instistute Ivrea ( nước Ý ).

Arduino là một nền tảng mà mọi thiết bị phần cứng đã được làm sẵn và chuẩn hoá, người dùng chỉ cần lựa chọn những thứ mình cần và lắp ghép lại. Chính vì thế, Arduino rất thích hợp cho những ai không có chuyên môn về điện tử nhưng lại yêu thích phần cứng – tự động hoá.

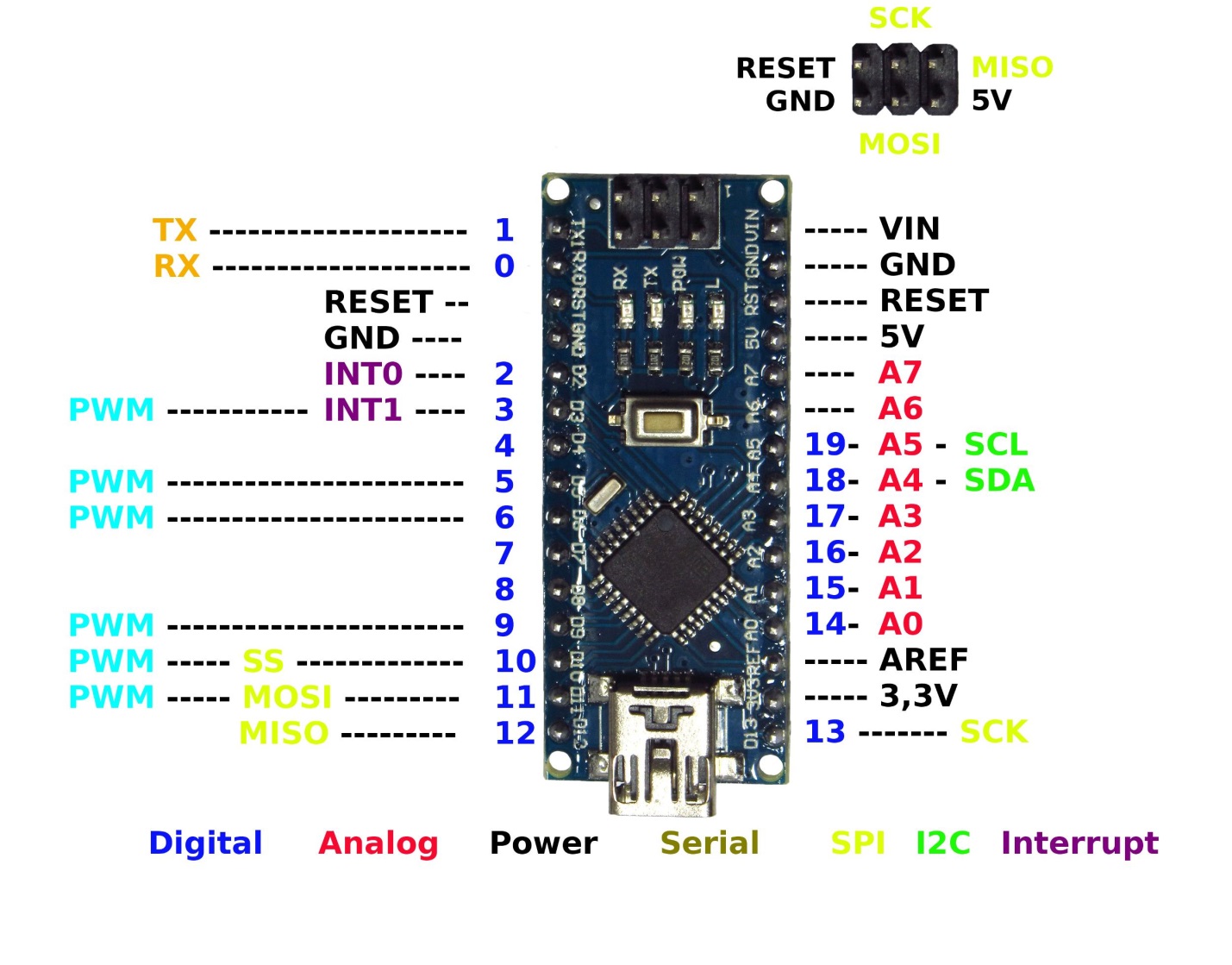
Tuy nhiên, để Arduino có thể hoạt động được thì chúng ta cần lập trình cho nó. Ngôn ngữ lập trình được sử dụng là C nhưng được đơn giản hoá đi nhiều để phù hợp với những người không có chuyên môn. Sự đơn giản hoá của Arduino chính là việc các phần cứng kết nối được với nó đều có các đoạn lệnh viết sẵn từ nhà sản xuất hoặc các nhà phát triển để có thể tự do sử dụng theo các mục đích khác nhau.



*Các bo mạch ArduinoMạch phát triển Arduino Nano*

Hiện nay, phiên bản Arduino Uno đã được phát triển đến thế hệ thứ 3 (Nano).

**Sơ đồ các khối**



*Cái khối chính của mạch Arduino Uno*

* Khối chuyển đổi Serial – USB: giao tiếp COM ảo qua cổng USB, nạp chương trình.
* Khối nguồn: cân bằng điện áp cho đúng với điện áp yêu cầu, cung cấp nguồn điện cho toàn hệ thống.

Các cổng điện áp: cung cấp điện áp đầu ra cho các thiết bị.

* Các cổng Analog in: cổng logic 0-1 vào với mức điện 0-5V tương ứng.
* Khối các cổng Digital in-out: có thể nhận điện áp vào hoặc xuất điện áp ra. Các cổng này có thể xuất xung PWM.

**Thông tin:**

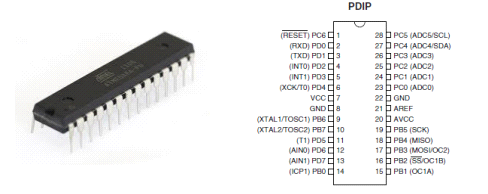
|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 |
| Điện áp hoạt động | 5V – DC |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | 30 mA |
| Điện áp khuyên dùng | 7-12V – DC |
| Điện áp giới hạn | 6-20V – DC |
| Số chân Digital | 14 ( 6 chân PWM) |
| Số chân Analog | 6 |
| Dòng tối đa mỗi chân I/O | 30 mA |
| Dòng ra tối đa chân 5V | 500 mA |
| Dòng ra tối đa chân 3.3V | 50 mA |
| Bộ nhớ Flash | 32KB (0.5 KB sử dụng làm bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |

*Thông tin chi tiết của Arduino Uno R3*

**Vi điều khiển**

Arduino Uno có thể sử dụng 3 vi kiều khiển họ 8bit là ATmega8, ATmega168, ATmega328. Điểm khác biệt của 3 vi điều khiển này là dung lượng bộ nhớ Flash.

Trong hệ thống này ta sử dụng ATmega328.



*Hình ảnh Atmega 328 và sơ đồ chân kết nối*

Đặc điểm:

* Cấu trúc RISC với khoảng 133 lệnh ASM, hầu hết thực hiện trong một chu

kỳ máy.

* 32 thanh ghi đa năng làm việc như các vùng nhớ tốc độ cao.
* Tốc độ thực hiện lệnh tối đa 16 MIPS (với thạch anh 16MHz, không chia

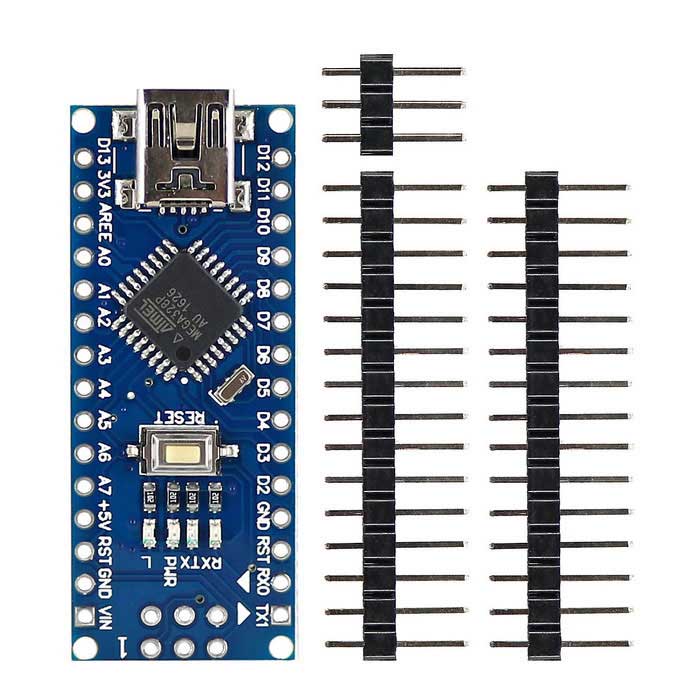
tần).

* 32KB bộ nhớ Flash, dùng làm bộ nhớ chương trình, cho phép nạp xoá.
* 1KBytes EEPROM, dùng để lưu các biến ngay cả khi không được cấp điện.
* 2KB SRAM dùng để lưu kết quả trung gian, làm bộ nhớ vào ra và dùng cho

stack.

* Giao diện JTAG, cho phép nạp xoá Flash 10.000 lần, EEPROM 100.000 lần, Fulse Bits, Lock Bits.
* Các chức năng ngoại vi tích hợp sẵn: 2 bộ Timer 8 bits với nhiều chế độ hoạt động, 2 bộ Timer 16 bits có bộ chia tần riêng, các chế độ đếm, định thời, so sánh (compare mode) và bắt giữ (capture mode).
* Bộ đồng hồ thời gian thực.
* 2 kênh điều chế độ rộng xung PWM có độ phân giải 8bit.
* 4 kênh điều chế độ rộng xung PWM có độ phân giải 2 đến 16 bits.
* Bộ ADC 10bit.
* Chuẩn giao tiếp TWI (I2C).
* Chuẩn truyền thông USART.
* Giao diện SPI master/slave.

**Các cổng vào ra**



*Các chân cắm của ArduinoUno Nano*

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).

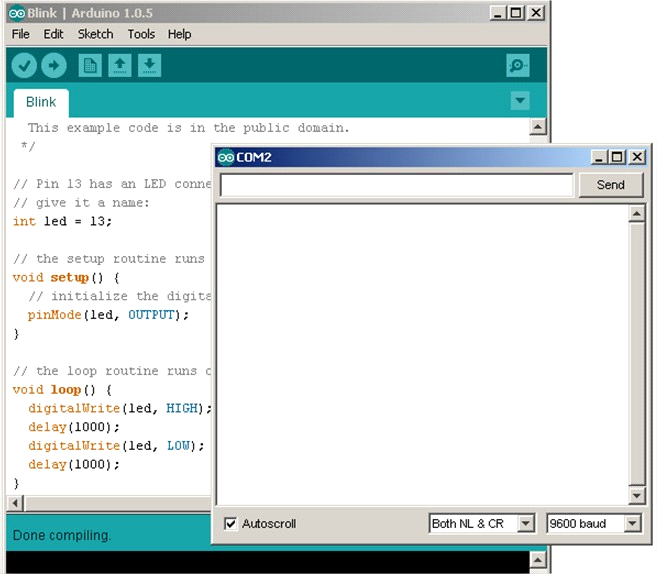
Một số chân digital có các chức năng đặc biệt như sau:

* 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 → 28-1 tương ứng với 0V → 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 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 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  → 2.5V với độ phân giải vẫn là 10bit.

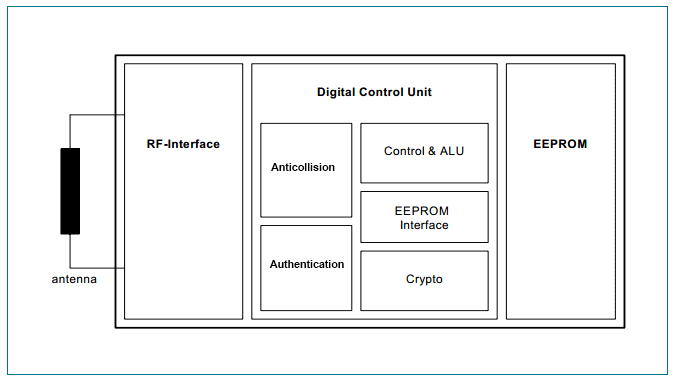
* **Lập trình cho Arduino**

Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cũng cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (**I**ntergrated **D**evelopment **E**nvironment)  như hình dưới đây.



*Arduino IDE*

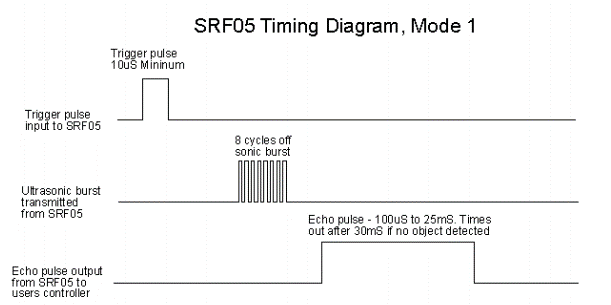
**Sơ đồ khối**



*Sơ đồ khối Xử Lý*

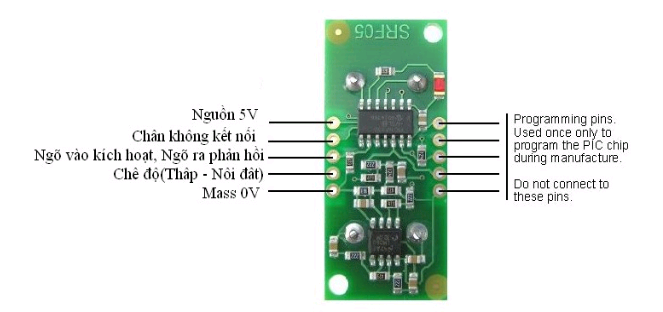
* RF Interface: điều chế/giải điều chế, chỉnh lưu, tạo xung đồng hồ, điều chỉnh điện áp.
* Anticollision: chọn và thực hiện field.
* Authentication: xác thực hoạt động truy cập.
* Control & ALU: tăng, giảm các giá trị đặc biệt.
* EEPROM Interface: quản lý các hoạt động liên quan tới EEPROM.
* Crypto: xác thực và mã hoá dữ liệu.
* EEPROM: lưu trữ dữ liệu.
* **Tổ chức dữ liệu**

1024 x 8 bit dữ liệu được tổ chức trong 16 sector với 4 khối, mỗi khối 16 byte. Ở trạng thái bị xoá, các cell EEPROM nhận giá trị logic 0, ở trạng thái được ghi, cell nhận giá chị logic 1.

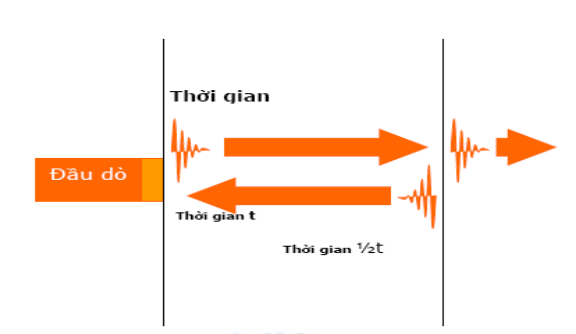
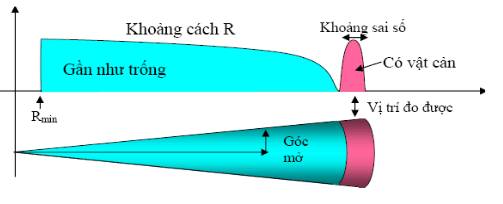


*Tổ chức dữ liệu*

* **Kết nối phần cứng**

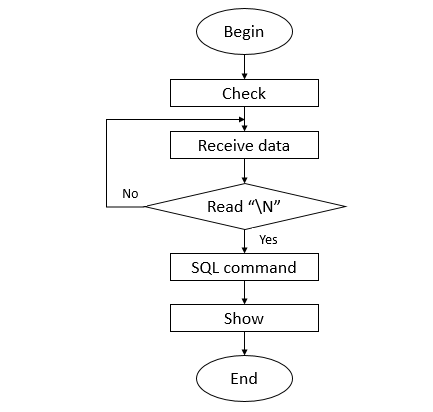


*Sơ đồ chân kết nối Arduino và MFRC522*

* **Hệ thống điều khiển**
* **Lưu đồ thuật toán trên bộ xử lý**
  + - 1. 
      2. 

*Lưu đồ thuật toán trên bộ vi xử lý*

* **Lưu đồ hệ thống phần mềm**



*Lưu đồ thuật toán hệ thống phần mềm*

* Bước 1: Khởi tạo kết nối với bộ vi xử lý.
* Bước 2: Kiểm tra trạng thái sóng
* Bước 3: Tiếp nhận dữ liệu từ bộ xử lý, nếu chưa gặp kí tư xuống dòng thì tiếp tục nhận dữ liệu.
* Bước 4: Khi gặp kí tự xuống dòng, tức là đã kết thúc chuỗi dữ liệu. Chương trình sẽ thực hiện truy vấn với đầu vào là chuỗi dữ liệu vừa nhận được.
* Bước 5: Hiển thị thông tin lên trên màn hình.
* Bước 6: Kết thúc phiên làm việc.
* **Code chương trình nạp mạch Arduino**

/ chuong trinh lap trinh arduino :

[#include](https://www.youtube.com/results?search_query=%23include) <LiquidCrystal.h>

[#include](https://www.youtube.com/results?search_query=%23include) <EEPROM.h> /

/---------------- lcd ------------------

LiquidCrystal lcd(12, 11, 5, 6, 9, 10);

// EN,RS,D4,D5,D6,D7 //

---------------- define pin ------------------

[#define](https://www.youtube.com/results?search_query=%23define) clk 4

[#define](https://www.youtube.com/results?search_query=%23define) data 3

[#define](https://www.youtube.com/results?search_query=%23define) loa A4

[#define](https://www.youtube.com/results?search_query=%23define) TRIG\_PIN 8

[#define](https://www.youtube.com/results?search_query=%23define) ECHO\_PIN 7

[#define](https://www.youtube.com/results?search_query=%23define) TIME\_OUT 5000 int A;

int a;

float biendo=100;

//---------------- valuable ------------------

int mode=0;

int mainmode=0;

long distance;

float GetDistance()

{ long duration,distanceCm; digitalWrite(TRIG\_PIN, LOW); delayMicroseconds(2); digitalWrite(TRIG\_PIN, HIGH); delayMicroseconds(10); digitalWrite(TRIG\_PIN, LOW); duration = pulseIn(ECHO\_PIN, HIGH, TIME\_OUT);

// convert to distance distanceCm = duration / 29.1 / 2; return distanceCm; }

void setup()

{

//---------------------------- Pinmode---------------------

/////////////////// input /////////////////////

pinMode(2,INPUT\_PULLUP);

pinMode(3,INPUT\_PULLUP);

pinMode(4,INPUT\_PULLUP);

pinMode(A2, OUTPUT);

pinMode(A3, OUTPUT);

pinMode(A4, OUTPUT);

pinMode(TRIG\_PIN, OUTPUT);

pinMode(ECHO\_PIN, INPUT);

//---------------------------- Interrupt --------------------- attachInterrupt(0, ngat, LOW);

//---------------------------- LCD ,

serial --------------------- lcd.begin (16,2);

Serial.begin(9600);

A= digitalRead(clk); }

void loop()

{

//-------------- dieu kien mode --------------------- i

f(mode==1)

{

detachInterrupt(0);

//ngat phan cung delay(50);

lcd.setCursor(0,0);

lcd.print("cam bien srf05");

mode=0; mainmode++;

if(mainmode>2) mainmode=0;

while(digitalRead(2)==0);

lcd.clear(); }

attachInterrupt(0, ngat, LOW);

//---------------------------- mode ---------------------

if(mainmode==0)

{lcd.setCursor(0,0);

lcd.print("CAM BIEN SRF05 ");

dokhoangcach(); }

if(mainmode==1)

{ up\_down();

chopbiendo(); }

if(mainmode==2)

{ lcd.setCursor(0,0);lcd.print("luu gia tri");

delay(1000);

mainmode=0; } }

//--------------------------------------- chuong trinh con -------------------------------------------

void up\_down() {

a =digitalRead(clk);

if(a != A)

{if(digitalRead(data) != a )

{ biendo = biendo + 1;

// chinh len if(biendo < 10) biendo = 10;

lcd.print(biendo); }

else {biendo = biendo - 1;

// chinh xuong if(biendo > 200) biendo = 200;

lcd.setCursor(5,1);

lcd.print(biendo); } } A=a; }

void chopbiendo()

{ lcd.setCursor(0,0);

lcd.print("khoang cach bao: ");

lcd.setCursor(5,1);

lcd.print(biendo);

lcd.setCursor(13,1);

lcd.print("cm");

lcd.print(" "); }

void ngat()

{ mode=1; pip\_1(); for(int j=0;j<100;j++); }

void pip\_1()

{ digitalWrite(loa,HIGH);

delay(50);

digitalWrite(loa,LOW);

delay(50); }

void dokhoangcach()

{long distance = GetDistance();

if (distance <= 0)

{ lcd.setCursor(0,1);

lcd.print("kh/cach : ");

lcd.print(" ");

lcd.print("cm");

lcd.print(" "); }

else { lcd.setCursor(0,1);

lcd.print("kh/cach : ");

lcd.print(distance);

lcd.print("cm");

lcd.print(" ");

if(distance<= biendo)

{pip\_1();} } delay(100); }﻿

* **Giao diện hệ thống**

**

* **KẾT LUẬN**
* Nghiên cứu và thiết kế thành công hệ thống chấm công tự động bằng thẻ từ ứng dụng công nghệ RFID

Gồm:

* Thiết bị đọc
* Bộ vi xử lý
* Chương trình hệ thống
* Ưu điểm: Thao tác dữ liệu nhanh, truy vấn dữ liệu chính xác, chương trình hệ thống đơn giản, dễ nhìn, dễ thao tác.
* Nhược điểm: Hệ thống phần cứng vẫn còn cồng kềnh, chương trình hệ thống mới chỉ giao tiếp được với một thiết bị đọc, còn chưa có hệ thống cửa vật lý kèm theo.
* Những điểm cần khắc phục và nâng cấp:
* Thay thế kết nối hữu tuyến bằng kết nối vô tuyến (wifi).
* Nâng cấp phần mềm hệ thống để có thể giao tiếp được với nhiều thiết bị đọc.
* Thiết kế và xây dựng thêm hệ thống cửa vật lý.
* Thiết kết phần cứng gọn gàng hơn.
* **TÀI LIỆU THAM KHẢO**

[1] Radio-frequency identification, Wikipedia

[2] [http://www.nxp.com](http://www.nxp.com/)

[3] [http://arduino.cc](http://arduino.cc/)

[4] [http://hocavr.com](http://hocavr.com/)