BỘ GIÁO DỤC & ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN – ĐIỆN TỬ

ĐỒ ÁN MÔN HỌC 1

NGÀNH KỸ THUẬT ĐIỆN-ĐIỆN TỬ

ĐỀ TÀI:

ĐO NHIỆT ĐỘ DÙNG CẢM BIẾN DS18B20 SỬ DỤNG BOARD ARDUINO, HIỂN THỊ TRÊN LCD, TRUYỀN PHÁT KHÔNG DÂY,GIAO TIẾP VỚI MÁY TÍNH QUA CỔNG COM

GVHD : NGUYỄN THANH BÌNH

SVTH : NGUYỄN VĂN QUỐC

MSSV : 11141170

Tp. Hồ Chí Minh - 5/2014

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG SỐ LIỆU

LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và vi điều khiển PIC ngày càng thông dụng và hoàn thiện hơn, nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử. Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Ngôn ngữ lập trình trên nền Java lại vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Trên cơ sở kiến thức đã học trong môn học : Tin học đại cương, vi xử lý 1 & 2, điện tử cơ bản, kỹ thuật số… cùng với những hiểu biết về các thiết bị điện tử, em đã quyết định thực hiện đề tài**: ĐO NHIỆT ĐỘ DÙNG CẢM BIẾN DS18B20 SỬ DỤNG BOARD ARDUINO, HIỂN THỊ TRÊN LCD, TRUYỀN PHÁT KHÔNG DÂY,GIAO TIẾP VỚI MÁY TÍNH QUA CỔNG COM** với mục đích để tìm hiểu thêm về Arduino, làm quen với các thiết bị điện tử,cách lập trình giao tiếp với máy tính và nâng cao hiểu biết cho bản thân. Do kiến thức còn hạn hẹp, thêm vào đó đây là lần đầu em thực hiện đồ án nên chắc chắn không tránh khỏi những thiếu sót, hạn chế vì thế em rất mong có được sự góp ý và nhắc nhỡ từ thầy giáo để có thể hoàn thiện đề tài của mình.

Em xin chân thành cảm ơn thầy giáo Nguyễn Thanh Bình đã giúp đỡ em rất nhiều trong quá trình tìm hiểu, thiết kế và hoàn thành đề tài đồ án 1 này.

TP HCM, ngày 25 tháng 05 năm 2014

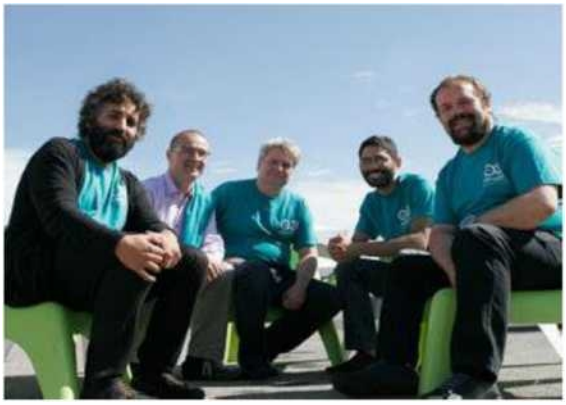
Sinh viên thực hiện

Nguyễn Văn Quốc

PHẦN 1
TỔNG QUAN

I. Giới thiệu chung về Arduino

Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY (là những người tự chế ra sản phẩm của mình) trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động, số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng phải ngạc nhiên về mức độ phổ biến.



Hình 1.1: Những thành viên khởi xướng Arduino.

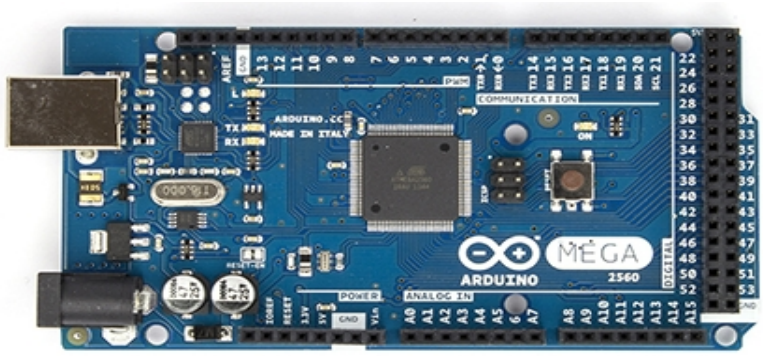
Arduino là gì mà có thể khiến ngay cả những sinh viên và nhà nghiên cứu tại các trường đại học danh tiếng như MIT, Stanford, Camegie Mellon phải sử dụng; hoặc ngay cả Google cũng muốn hỗ trợ khi cho ra đời bộ kit Arduino Mega ADK dùng để phát triển các ứng dụng Android tương tác với cảm biến và các thiết bị khác.

Arduino thật ra 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 cực kỳ 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 hiện tượng 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 board Arduino có 20 ngõ I/O có thể tương tác và điều khiển chừng ấy 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 Institute 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.

II. Giới thiệu về board Arduino Mega 2560



Hình 1.2: Board Arduino Mega 2560.

Arduino Mega 2560 là 1 bo mạch thiết kế với bộ xử lý trung tâm là vi điều khiển AVR Atmega2560. Cấu tạo chính của Arduino Mega 2560 bao gồm các phần sau:

- *Cổng USB:* đây là loại cổng giao tiếp để ta upload code từ PC lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển và máy tính.
- *Jack nguồn*: để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên, nhưng không phải lúc nào cũng có thể cắm với máy tính được. Lúc đó ta cần một nguồn từ 9V đến 12V.
- Có 54 chân vào/ra số đánh số thứ tự từ 0 đến 13, ngoài ra có một chân

nối đất (GND) và một chân điện áp tham chiếu (AREF).

- *Vi điều khiển AVR:* đây là bộ xử lí trung tâm của toàn bo mạch. Với mỗi mẫu Arduino khác nhau thì con chip là khác nhau. Ở con Arduino Mega2560 này thì sử dụng ATMega2560.
- Các thông số chi tiết của Ardiuno Mega 2560:

Vi xử lý:	5V
Điện áp hoạt động:	7-12V
Điện áp đầu vào:	6-20V
Chân vào/ra (I/O) số:	5 4 (1 5 chân là đầu ra PWM)
Chân vào tương tự:	16
Dòng điện trong mỗi chân I/O:	40mA
Dòng điện Chân nguồn 3.3V:	50mA
Bộ nhớ trong:	256 KB
SRAM:	8 KB
EEPROM:	4 KB
Xung nhịp:	16MHz
Atmega2560	

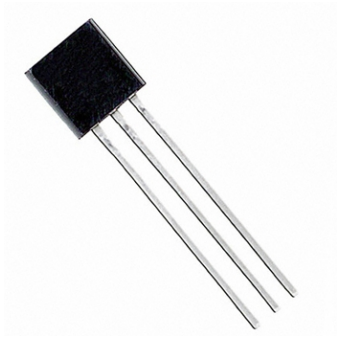
Các Mega 2560 có 16 đầu vào tương tự, mỗi ngõ vào tương tự đều có độ phân giải 10 bit (tức là 1024 giá trị khác nhau).Theo mặc định đo từ 0 đến 5 volts, mặc dù là nó có thể thay đổi phần trên của phạm vi bằng cách sử dụng chân Aref và analogReference) chức năng.

Các Atmega 2560 có 256 KB bộ nhớ flash để lưu trữ mã (trong đó có 8 KB được sử dụng cho bộ nạp khởi động), 8 KB SRAM và 4 KB của EEPROM.

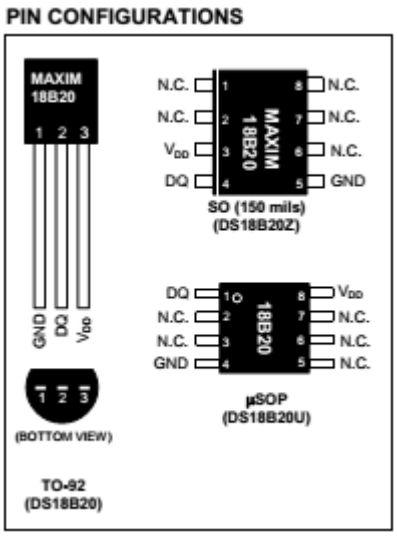
III. Giới thiệu về cảm biến nhiệt độ DS18B20

1.Tổng quan:

DS18B20 là IC cảm biến nhiệt độ, chỉ bao gồm 3 chân,hình ảnh thức tế như hình dưới.



Hình 1.3: Cảm biến DS18B20



Hình 1.4: Sơ đồ chân cảm biến DS18B20.

2.Đặc điểm DS18B20:

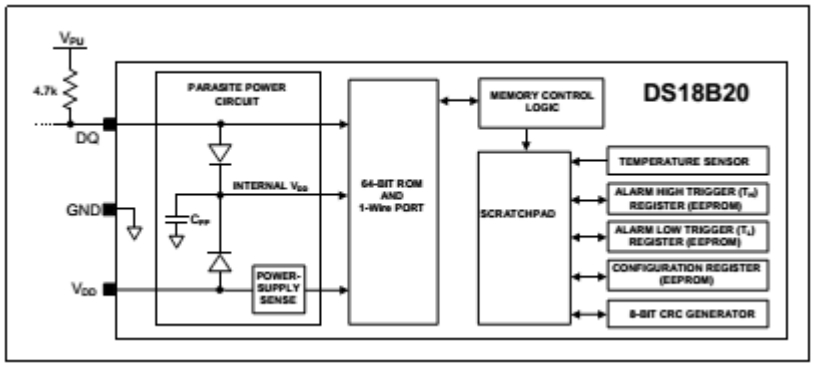
- IC đo nhiệt độ,giao tiếp với VDK qua giao thức 1 dây.
- Mỗi thiết bị có 1 mã code 64 bit riêng biệt.
- Nguồn cung cấp 3V-5.5V,có thể cấp nguồn thông qua chân dữ liệu.
- Có thể đo được khoảng nhiệt độ từ -55ºC đến +125ºC.

– Độ chính xác 0.5°C trong khoảng nhiệt độ đo từ -10°C đến 85°C.

– Độ phân giải cảm biến 9-12 bit.

– Thời gian chuyển đổi lớn nhất 750ms tương ứng với độ phân giải 12bit.

Sơ đồ khối bên trong của cảm biến:



Hình 1.5: Sơ đồ khối DS18B20.

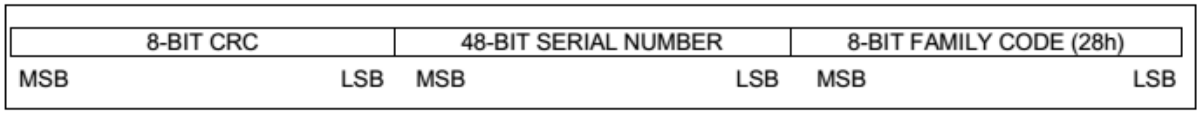
3.Giao tiếp với DS18B20:

– VDK giao tiếp với DS18B20 theo từng chu kì.

– Mỗi lần truy xuất dữ liệu từ DS18B20 phải trãi qua 3 bước:

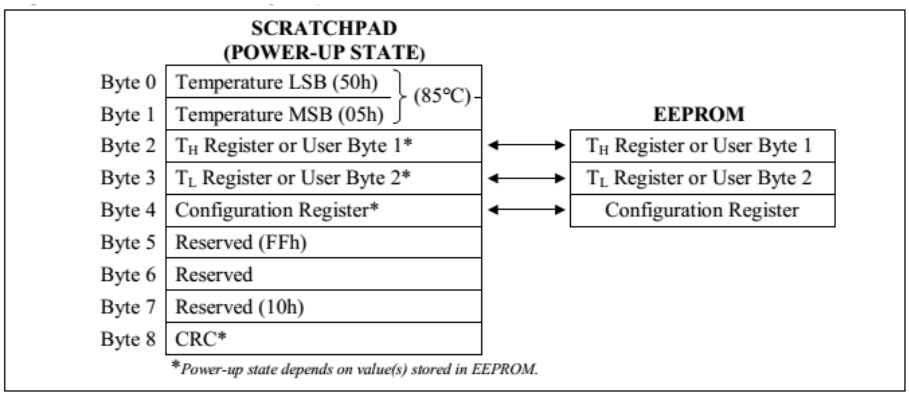
- Bước 1:Khởi tạo
- Bước 2:Gửi mã lệnh ROM
- Bước 3:Gửi lệnh chức năng cho DS18B20 thực hiện

– Cấu trúc vùng nhớ mã ROM 64 bit của DS18B20:



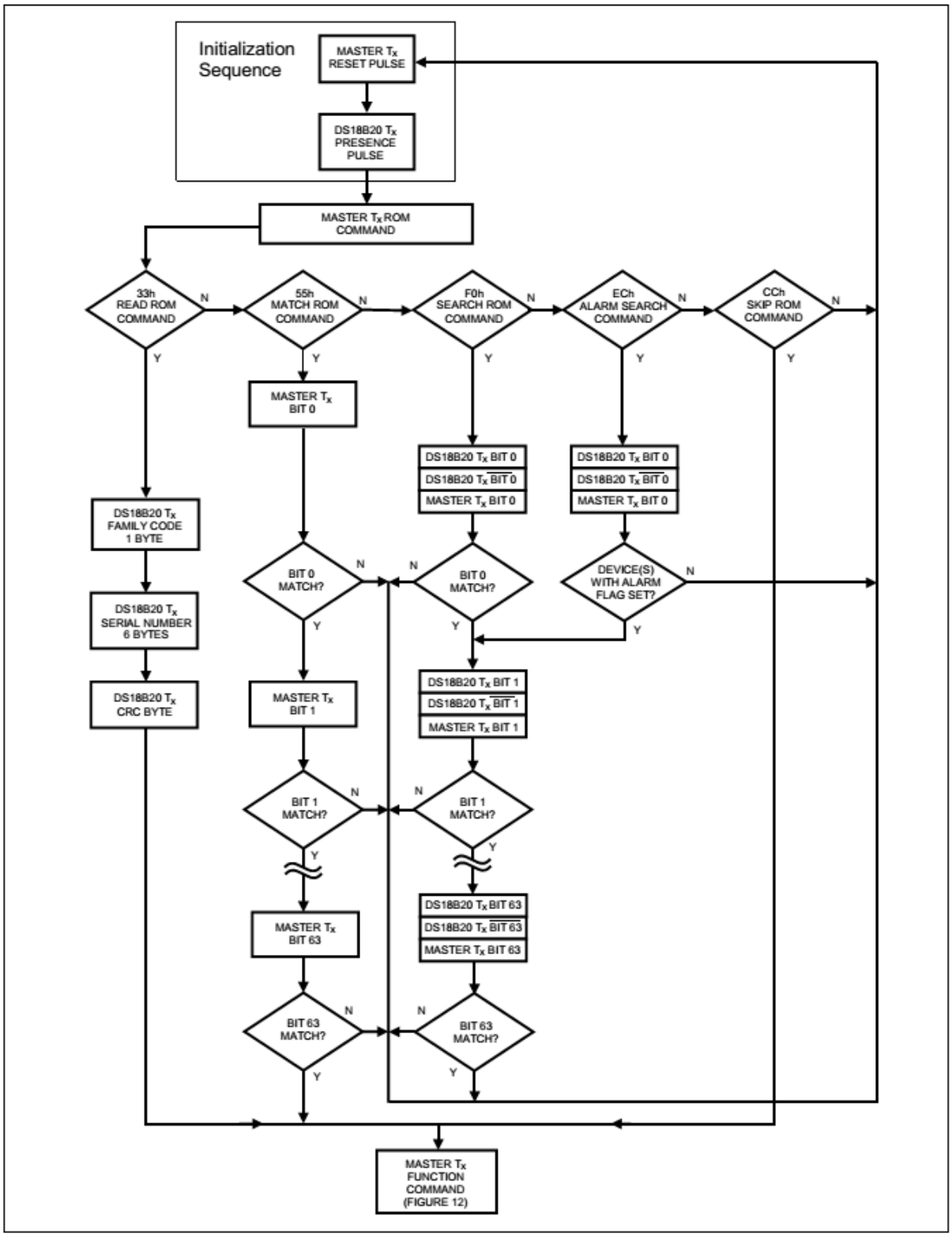
Hình 1.6: Mã 64bit mã ROM

– Sơ đồ vùng nhớ DS18B20:



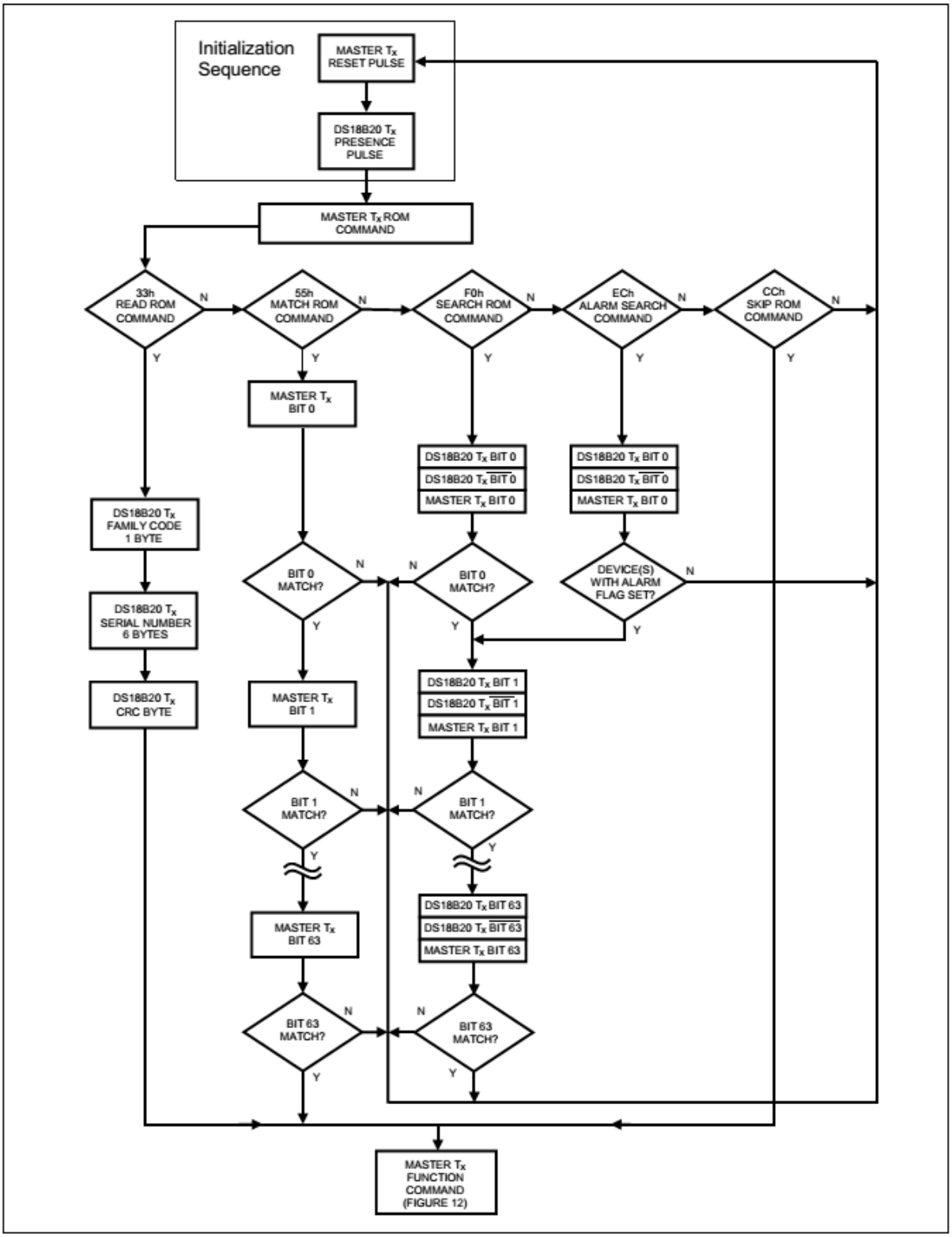
Hình 1.7: Cấu trúc vùng nhớ DS18B20

– Lưu đồ lệnh ROM DS18B20 được trình bày bên dưới:



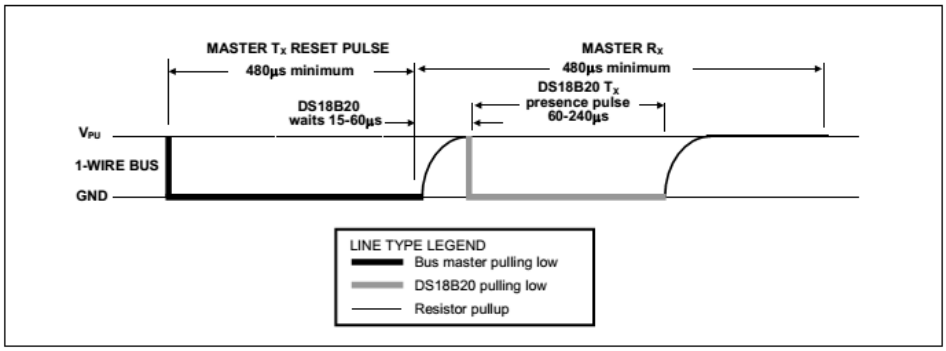
Hình 1.8: Lưu đồ lệnh ROM

- Lưu đồ lệnh chức năng DS18B20 được trình bày bên dưới:



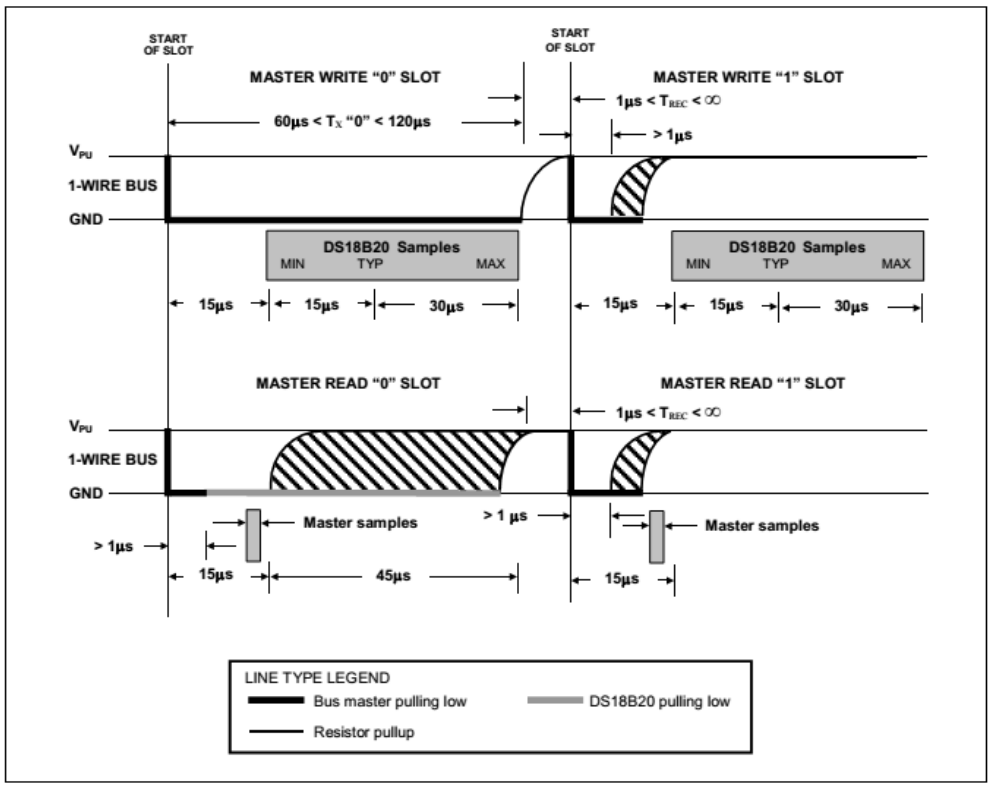
Hình 1.9: Lưu đồ lệnh chức năng DS18B20

– Thời gian khởi tạo:



Hình 1.10: Khe thời gian khởi tạo

– Giản đồ khe thời gian đọc viết:



Hình 1.11: Khe thời gian đọc,viết

IV. Giới thiệu về LCD 16x2:

– Thông số kỹ thuật:

Hình 1.12: Hình ảnh sơ đồ chân LCD 16x2.

– Lcd có tất cả 16 chân:

- Chân cấp nguồn: vss (nối nguồn 5V), VDD (nối 0V), V0 (điều chỉnh độ tương phản)

- RS: Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi.

+ Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read).

+ Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.

- RW: Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.

- E: Chân cho phép chốt xung kí tự (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.

+Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.

+ Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.

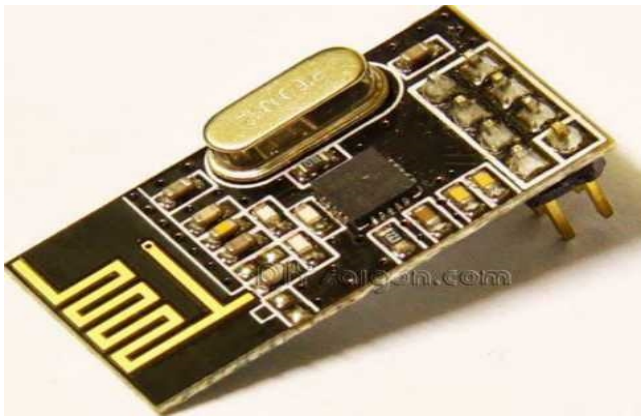
-D0-D7: Chân dữ liệu

-A, K: Chân điều khiển đèn nền

Lcd có thể hoạt động theo 2 chế độ: 4 bit và 8 bit. Chế độ 4 bit đòi hỏi phải kết nối với 7 chân I/O của ardiuno. Chế độ 8 bit đòi hỏi phải kết nối với 11 chân I/O của Ardiuo. Trong đề tài này em chọn LCD hoạt động ở chế độ 4 bit

v. Giới thiệu về module truyền phát nRF24L01

1. Thông số kỹ thuật



Hình 1.13: Module nRF24L01.

-Radio:

+Hoạt động ở giải tần 2.4Ghz

+ Có 126 kênh

+Truyền và nhận dữ liệu

+ Truyền tốc độ cao lMbps hoặc 2Mbps.

- Công suất phát:

+ Có thể cài đặt được 4 công suất nguồn phát: 0,-6,-12,-18dBm

- Thu:

+ Có bộ lọc nhiễu tại đầu thu

+ Khuếch đại bị ảnh hưởng bởi nhiễu thấp (LNA)

-Nguồn cấp:

+ Hoạt động từ 1.9-3.6V

+ Các chân IO chạy được cả 3.3 lẫn 5V

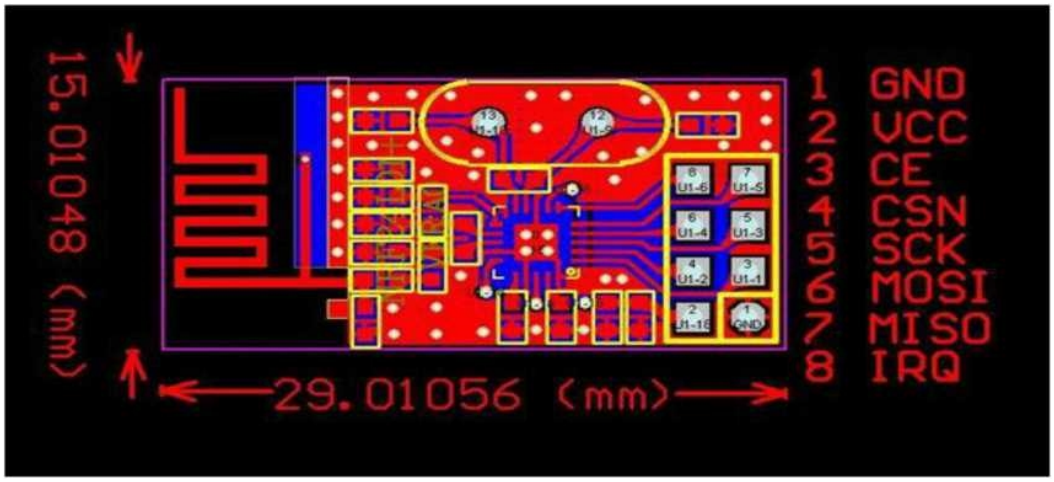
- Giao tiếp:

+ 4 chân giao tiếp theo giao thức SPI

+ Tốc độ tối đa 8Mbps

+ 3-32 bytes trên 1 khung truyền nhận.

2. Sơ đồ chân:



Hình 1.14: Sơ đồ chân module nRF24L01.

3. Phân tích

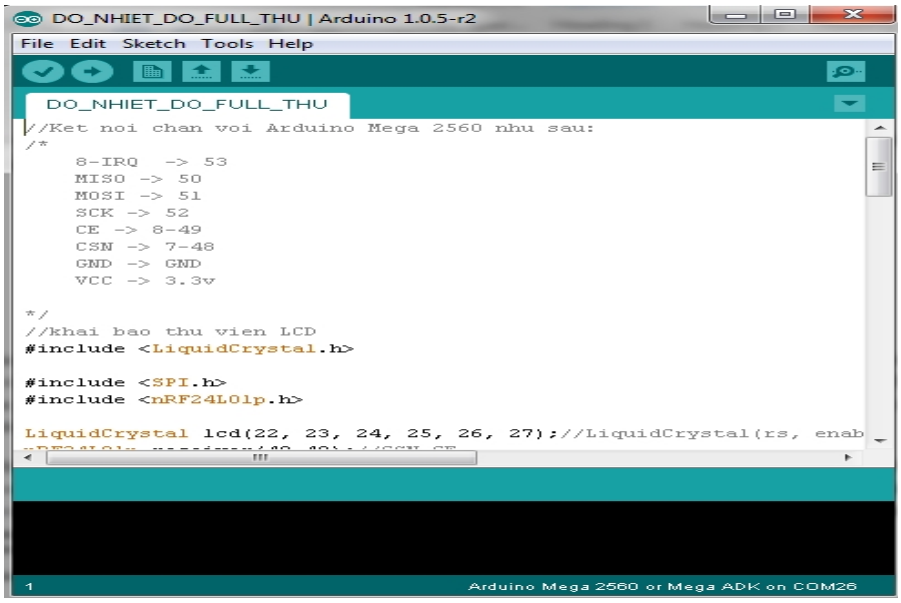
+Module nRF24L01 hoạt động ở tần số sóng ngắn 2.4G nên Module này khả năng truyền dữ liệu tốc độ cao và truyền nhận dữ liệu trong điều kiện môi trường có vật cản.

+Module nRF24L01 có 126 kênh truyền. Điều này giúp ta có thể truyền nhận dữ liệu trên nhiều kênh khác nhau.

+Module khả năng thay đổi công suất phát bằng chương trình, điều này giúp nó có thể hoạt động trong chế độ tiết kiệm năng lượng.

+ Chú ý: Điện áp cung cấp cho là 1.9V đến 3.6V. Điện áp thường cung cấp là 3.3V. Nhưng các chân 10 tương thích với chuẩn 5V. Điều này giúp nó giao tiếp rộng dãi với các dòng vi điều khiển.

VI. Giới thiệu về ngôn ngữ lập trình cho Arduino.



Hình 1.15: Giao diện phần mềm Arduino IDE.

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 Wiring 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

Arduino IDE là phần mềm dùng để lập trình cho 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 vào chương trình nếu muốn.Hiện tại, Arduino IDE có thể download từ trang chủ http://arduino.cc/ bao gồm các phiên bản sau:

- Arduino 1.0.5

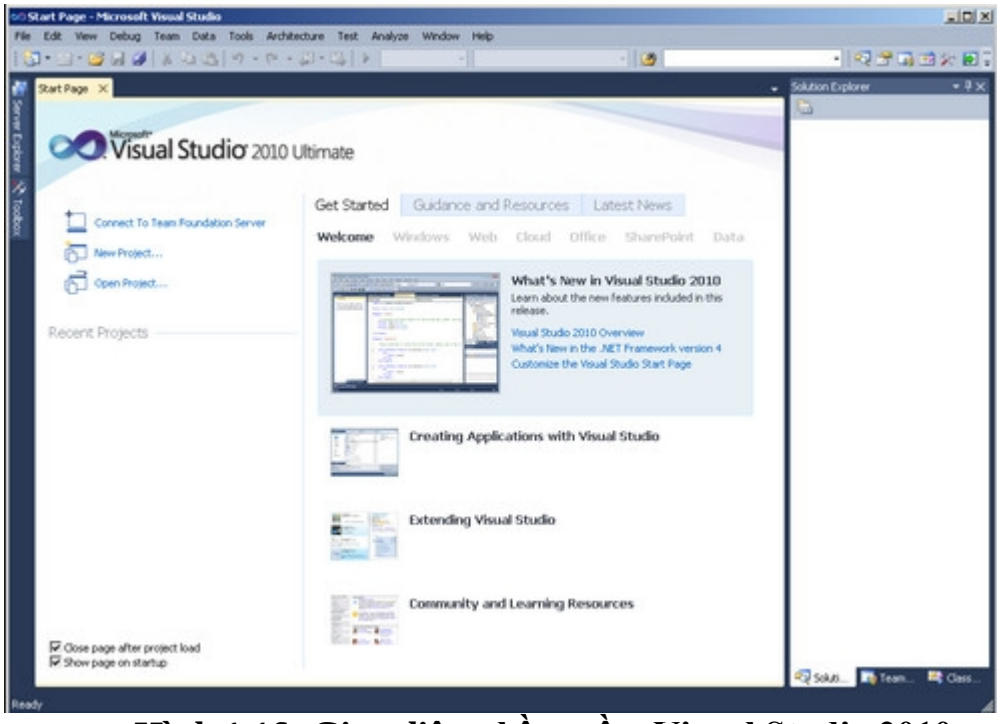
-Arduinol.5.5 BETA (Hỗ trợ cho 2 board Arduino mới nhất là: Arduino Yun và Arduino Due).

-Arduino IDE cho Intel Galileo.

VII.Giới thiệu phần mềm Visual Studio 2010:

1.Tổng quan:

- Visual Studio 2010 Ultimate là công cụ xây dựng , lập trình mã nguồn để quản trị thông tin hệ thống phát triển phần mềm của doanh nghiệp, xây dựng các ứng dụng cho máy để bàn và các ứng dụng web.
- Visual Studio 2010 Ultimate được xem là một trong những công cụ thiết kế tốt nhất hiện nay với việc phát triển phần mềm, triển khai các giải pháp doanh nghiệp.



Hình 1.16: Giao diện phần mềm Visual Studio 2010.

- Visual Studio 2010 Ultimate được tăng cường thêm những giải pháp giảm thiểu nguy cơ trong quá trình phát triển thiết kế.
- Visual Studio 2010 Ultimate tạo ra các giải pháp về phần mềm, phát triển một số công cụ tuyệt vời của ứng dụng lập trình.

- Có thể nói Visual Studio 2010 Ultimate là phần mềm không thể thiếu dành cho những Kỹ thuật viên phần mềm và một số công ty phát triển phần mềm.

Những tính năng chính của Visual Studio 2010 Ultimate:

- Phát triển mã nguồn, phần mềm.
- Thực hiện giải pháp phần mềm.
- Ứng dụng lập trình.

2.Giới thiệu Windown Form C#:

2.1 Giới thiệu:

- Tạo ra các ứng dụng chạy trên máy tính có cài đặt .NET Framework 2.0.
- Sử dụng không gian System.Windows.Forms.
- Thiết kế giao diện trực quan.
- Ví dụ ta thiết kế giao diện như hình dưới:

Hình 1.17: Một giao diện đăng nhập do người dùng thiết kế.

2.2. Ứng dụng của Windowns Forms:

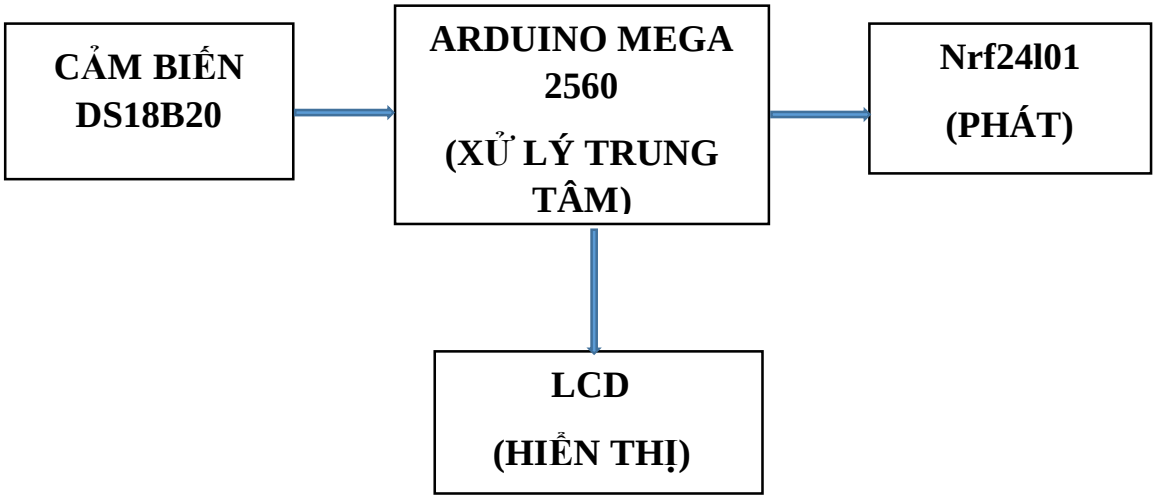
- Các chương trình quản lý tài chính dân sự,sản xuất,quản lý doanh nghiệp,..

PHẦN 2

LẬP TRÌNH VÀ LẮP ĐẶT MẠCH ĐO NHIỆT ĐỘ VÀ TRUYỀN PHÁT KHÔNG DÂY,VIẾT GIAO DIỆN NHẬN NHIỆT ĐỘ DÙNG C#

I. Mạch phát:

1. Sơ đồ các khối:



2. Chức năng các khối:

- Cảm biến: có chức năng đo nhiệt độ từ môi trường và gửi giá trị đo được cho Ardiuno khi có tín hiệu yêu cầu.
- Khối xử lý trung tâm: có chức năng điều khiển cảm biến DS18B20 đo nhiệt độ, đồng thời hiển thị dữ liệu trên LCD và điều khiển phát dữ liệu thông qua module NRF24L01.
- Khối hiển thị: có chức năng hiển thị giá tri nhiệt độ đo được.
- Khối phát: có chức năng phát dữ liệu(nhiệt độ đo được) từ Ardiuno này sang bộ thu của Arduino khác.

3. Sơ đồ kết nối phần cứng:

a. Bảng 1. Bảng kết nối chân Arduino với LCD:

Arduino pin	22	23	24	25	26	27
LCD pin	RS	E	D4	D5	D6	D7

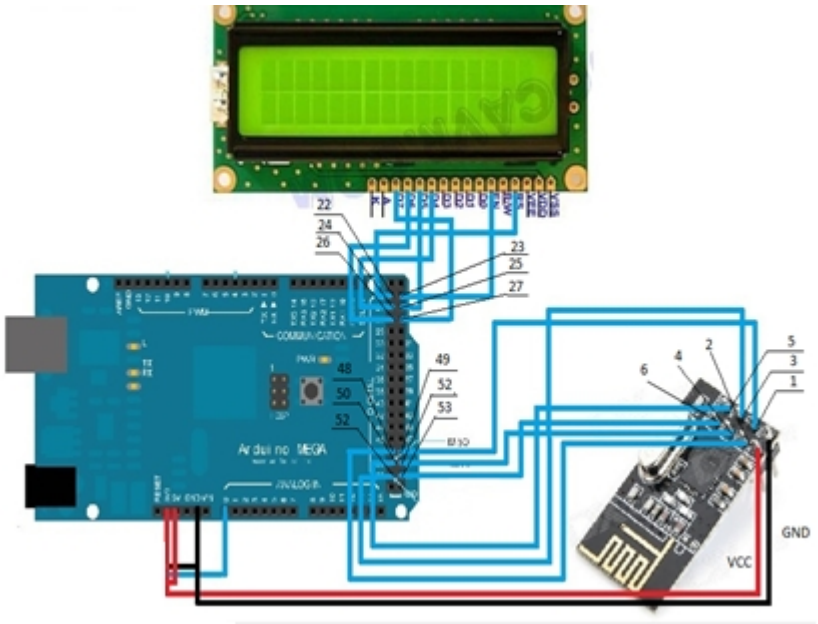
b. Bảng 2. Bảng kết nối chân Arduino với DS18B20:

Arduino pin	2 (ngõ vào số)	5V	GND
DS18B20	Vout	Vs	GND

c. Bảng 3. Bảng kết nối chân Arduino với nRF24L01:

Arduino pin	GND	3.3V	8	7	52	51	50	53
NRF24L01 PIN	GND	VCC	CE	CSN	SCK	MOSI	MISO	IRQ

d. Sơ đồ kết nối chung cho các khối:



Hình 1.18: Sơ đồ kết nối phần cứng bên phát.

4. Lập trình cho Arduino mạch phát:

4.1. Giới thiệu thư viện sử dụng trong chương trình đo và phát nhiệt độ:

+ Thư viện: module nRF24L01

```
#include <nRF24L01p.h>
```

+Thư viện: truyền dữ liệu

```
#include <SPI.h>
```

+ Thư viện: LCD

```
#include <LiquidCrystal.h>
```

+Thư viện :giao tiếp với cảm biến one wire

```
#include <OneWire.h>
```

4.2. CODE lập trình:

```
//khai bao thu vien LCD
#include <LiquidCrystal.h>
LiquidCrystal lcd(22, 23, 24, 25, 26, 27);
#include <SPI.h>
#include <nRF24L01p.h>
nRF24L01p transmitter(48,49);
#include <OneWire.h>
OneWire ds(2);
// Scratchpad locations
#define TEMP_LSB        0
#define TEMP_MSB        1
#define HIGH_ALARM_TEMP 2
#define LOW_ALARM_TEMP  3
#define CONFIGURATION   4
#define INTERNAL_BYTE   5
#define COUNT_REMAIN    6
#define COUNT_PER_C     7
#define SCRATCHPAD_CRC  8
```

```
// Device resolution
#define TEMP_9_BIT  0x1F //  9 bit
#define TEMP_10_BIT 0x3F // 10 bit
#define TEMP_11_BIT 0x5F // 11 bit
#define TEMP_12_BIT 0x7F // 12 bit
// OneWire commands
#define STARTCONVO      0x44  // Tells device to take a temperature
reading and put it on the scratchpad
#define COPYSCRATCH     0x48  // Copy EEPROM
#define READSCRATCH     0xBE  // Read EEPROM
#define WRITESCRATCH    0x4E  // Write to EEPROM
#define RECALLSCRATCH   0xB8  // Reload from last known
#define READPOWERSUPPLY 0xB4  // Determine if device needs
parasite power
#define ALARMSEARCH     0xEC  // Query bus for devices with an
alarm condition
int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;
int i;
float T;//nhiet do o gia tri do C
byte data[12];//mang de luu du lieu vung nho
byte addr[8],data_scratch[9];//mang de luu gia tri cua vung nho ROM
byte numberDevice,bit_resolution;
//ham xuat ra ki tu do C
byte ki_tu_do[8] = {
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
```

```
  B00000,
  B00000,
};
void countDevice()//function count number device 1-Wire on bus
{
  numberDevice=0;
  while(ds.search(addr))
  {
    numberDevice++;
  }
  Serial.print("Found : ");
  Serial.print(numberDevice);
  Serial.println(" device.");
}

bool get_Address(unsigned char* deviceAddress, byte index)
/*
ham tim du lieu vung nho ROM cua thiet bi thu index va
luu du lieu vao mang deviceAddress
*/
{
  byte index_temp=0;
  while(index_temp <= index && ds.search(deviceAddress))
  {
    if((index_temp == index) && (ds.crc8(deviceAddress, 7) ==
deviceAddress[7]))
        return true;
    index_temp++;
  }
  return false;
}
```

```
void display_ROM(byte n)//functon display memory ROM of all device
{
  int i,j;
  for(i=0;i<n;i++)
  {
    get_Address(addr, i);
    Serial.print("Found device ");
    Serial.print(i, DEC);
    Serial.println(" with : ");

    Serial.print("ROM memory : ");
    for( j = 7; j >=0; j--)
    {
    Serial.print(addr[j], HEX);
    Serial.print(" ");
    }
    Serial.println();
    //xac dinh ho thiet bi
    if ( addr[0] == 0x10)
    {
      Serial.println("Family device is : DS18S20");
    }
    else if ( addr[0] == 0x28)
    {
      Serial.println("Family device is : DS18B20");
    }
    else
    {
      Serial.print("Device family is not recognized: 0x");
      Serial.println(addr[0],HEX);
      return;
```

```
    }
    //cai dat do phan giai
    set_Resolution(addr, 12);
    //doc do phan giai
    get_Resolution(addr);
    Serial.print("Number bit Resoluton : ");
    Serial.println(bit_resolution);
  }
}
void read_Scratch(unsigned char* deviceAddress, unsigned char*
scratchPad)
{
  ds.reset();
  ds.select(deviceAddress);
  ds.write(READSCRATCH);
  // byte 0: temperature LSB
  scratchPad[TEMP_LSB] = ds.read();

  // byte 1: temperature MSB
  scratchPad[TEMP_MSB] = ds.read();

  // byte 2: high alarm temp
  scratchPad[HIGH_ALARM_TEMP] = ds.read();

  // byte 3: low alarm temp
  scratchPad[LOW_ALARM_TEMP] = ds.read();

  // byte 4:
  // DS18S20: store for crc
  // DS18B20 & DS1822: configuration register
  scratchPad[CONFIGURATION] = ds.read();
```

```
  // byte 5:
  // internal use & crc
  scratchPad[INTERNAL_BYTE] = ds.read();

  // byte 6:
  // DS18S20: COUNT_REMAIN
  // DS18B20 & DS1822: store for crc
  scratchPad[COUNT_REMAIN] = ds.read();

  // byte 7:
  // DS18S20: COUNT_PER_C
  // DS18B20 & DS1822: store for crc
  scratchPad[COUNT_PER_C] = ds.read();

  // byte 8:
  // SCTRACHPAD_CRC
  scratchPad[SCRATCHPAD_CRC] = ds.read();

  ds.reset();
}
void get_Resolution(unsigned char* deviceAddress)
{
  bit_resolution =0;
  read_Scratch(deviceAddress, data_scratch);
  if(data_scratch[CONFIGURATION]== 0x1F)
    {
     bit_resolution = 9;
    }
  else if(data_scratch[CONFIGURATION]== 0x3F)
    {
```

```
      bit_resolution = 10;
    }
  else if(data_scratch[CONFIGURATION]== 0x5F)
    {
      bit_resolution = 11;
    }
  else if(data_scratch[CONFIGURATION]== 0x7F)
    {
      bit_resolution = 12;
    }
}
void write_ScratchPad(unsigned char* deviceAddress, unsigned char*
scratchPad)
{
  ds.reset();
  ds.select(deviceAddress);
  ds.write(WRITESCRATCH);
  ds.write(scratchPad[HIGH_ALARM_TEMP]); // high alarm temp
  ds.write(scratchPad[LOW_ALARM_TEMP]); // low alarm temp
  ds.write(scratchPad[CONFIGURATION]); // configuration
  ds.reset();
  ds.write(COPYSCRATCH);
  delay(10); // 10ms delay
  ds.reset();
}
void set_Resolution(unsigned char* deviceAddress, byte newResolution)
{
  read_Scratch(deviceAddress, data_scratch);
  switch (newResolution)
     {
       case 12:
```

```
            data_scratch[CONFIGURATION] = TEMP_12_BIT;
            break;
          case 11:
            data_scratch[CONFIGURATION] = TEMP_11_BIT;
            break;
          case 10:
            data_scratch[CONFIGURATION] = TEMP_10_BIT;
            break;
          case 9:
          default:
            data_scratch[CONFIGURATION] = TEMP_9_BIT;
            break;
        }
        write_ScratchPad(deviceAddress, data_scratch);

}
void measure_T(byte n)//ham do nhiet do tat ca cac kenh
{
 int a,b;
 for(a=0;a<n;a++)
 {
   get_Address(addr, a);
   ds.reset();//khoi tao cho vong giao tiep ke tiep
   ds.select(addr);
   ds.write(0x44,1);         // DS18B20 Function Command:start conversion
Temp, with parasite power on at the end
   delay(1000);     // maybe 750ms is enough, maybe not
  // we might do a ds.depower() here, but the reset will take care of it.
   ds.reset();
   ds.select(addr);//MATCH ROM COMMAND
   ds.write(0xBE);         //DS18B20 Function Command: Read Scratchpad
```

```
  LowByte = ds.read();
  HighByte = ds.read();
  TReading = (HighByte << 8) + LowByte;
 /*ghep byte cao va byte thap cua thanh ghi nhiet do
 de duoc du lieu nhiet do 12 bit
 */
 SignBit = TReading & 0x8000;  // test most sig bit
 //ket qua tra ve la 1 neu bit thu 12 cua TReading la 1,nghia la nhiet do
am
 if (SignBit) // negative
 {
  TReading = (TReading ^ 0xffff) + 1; // 2's comp
 }
 Tc_100 = (6 * TReading) + TReading / 4;    // multiply by (100 *
0.0625) or 6.25

 Whole = Tc_100/ 100;  // separate off the whole and fractional portions
 Fract = Tc_100 % 100;
 T=Tc_100;
 T=T/100;

 //truyen du lieu nhiet do ra cong COM
  Serial.print("Temperature device ");
  Serial.print(a, DEC);
  Serial.println(" is : ");
  if (SignBit) // If its negative
    {
      Serial.print("-");
    }
  Serial.println(T);
```

```
    //hien thi nhiet do len LCD
    lcd.setCursor ( 9, a );
    lcd.print(T);

    //truyen nhiet do
    transmitter.txPL(a);
    transmitter.send(SLOW);
    transmitter.txPL(Tc_100);
    transmitter.send(SLOW);
   }
}
void setup(void) {
  lcd.createChar(0, ki_tu_do);//tao ki tu do "0"  ki tu do co ten "char(0)"
  //khoi tao LCD
  lcd.begin(16,2);//hien thi LCD 16 cot 2 hang
  //hien thi hang 1
  lcd.setCursor(0, 0);//lcd.setCursor(col, row) di chuyen con tro tai cot 0
hang 0
  lcd.print("Chanel 1:");

  //xuat ki tu char(0) ra LCD
  lcd.setCursor ( 14, 0 );
  lcd.print(char(0));

  //xuat ki tu "C" ra LCD
  lcd.setCursor(15,0);
  lcd.print("C");
  //hien thi hang 2
  lcd.setCursor(0, 1);//lcd.setCursor(col, row) di chuyen con tro tai cot 0
hang 0
  lcd.print("Chanel 2:");
```

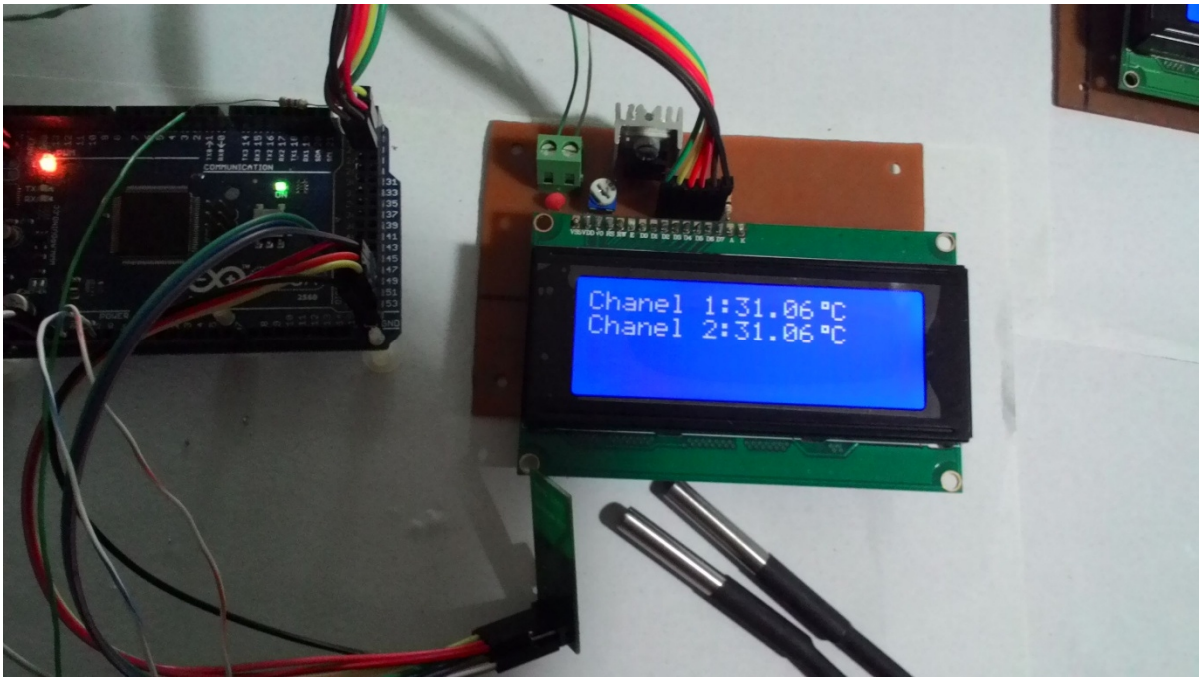
```
  //xuat ki tu char(0) ra LCD
  lcd.setCursor ( 14, 1 );
  lcd.print(char(0));

  //xuat ki tu "C" ra LCD
  lcd.setCursor(15,1);
  lcd.print("C");


  // start serial port
  Serial.begin(9600);
  //khoi nrf24l01
  SPI.begin();//khoi tao truyen du lieu kieu SPI
  transmitter.channel(10);//chon kenh truyen cho module nRF24L01
  transmitter.TXaddress("Artur");
  transmitter.init();
  //dem so thiet bi tren bus data
  countDevice();
  //
  display_ROM(numberDevice);

    get_Address(addr, 0);
    ds.reset();//khoi tao cho vong giao tiep ke tiep
    read_Scratch(addr, data_scratch);
}

void loop(void)
{
  measure_T(numberDevice);
  }
```

4.3. Nạp code và chạy chương trình:

Sau khi nạp chương trình cho Arduino thì hệ thống bắt đầu làm việc.

Hệ thống hoạt động theo trình tự như sau:

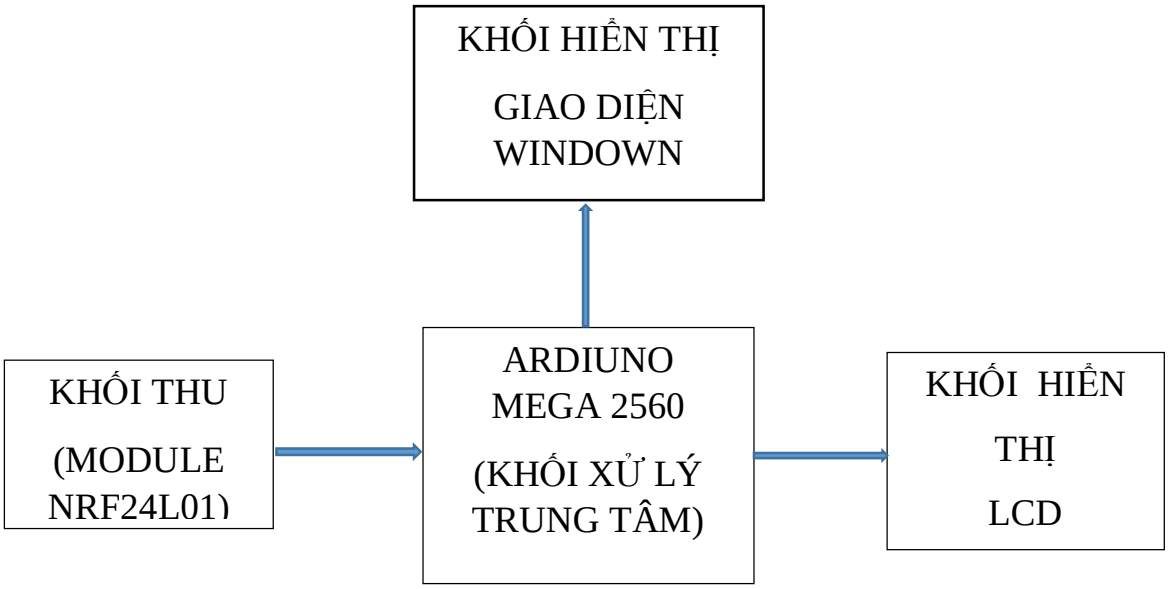
- Arduino điều khiển DS18B20 đo nhiệt độ và gửi lại nhiệt độ về arduino,việc giao tiếp xảy ra tại chân digital 2.
- Arduino gửi nhiệt độ đo được cho LCD 16x2 hiển thị.
- Arduino truyền nhiệt độ nhận được đến khối thu thông qua module nRF24L01.



Hình 1.19: Hình ảnh thực tế kết quả nhiệt độ bên phát.

II. MẠCH THU

1. Sơ đồ khối:



2. Chức năng các khối:

- Khối thu: có chức năng nhận dữ liệu nhiệt từ module NRF24L01 của máy phát.
- Khối xử lý trung tâm: có chức năng điều khiển nhận dữ liệu thông qua module NRF24L01, đồng thời hiển thị dữ liệu trên LCD.
- Khối hiển thị LCD: có chức năng hiển thị giá tri nhiệt độ thu được.
- Khối hiển thị trên giao diện Windowns:hiển thị nhiệt độ nhận được từ Arduino.

3. Sơ đồ kết nối phần cứng:

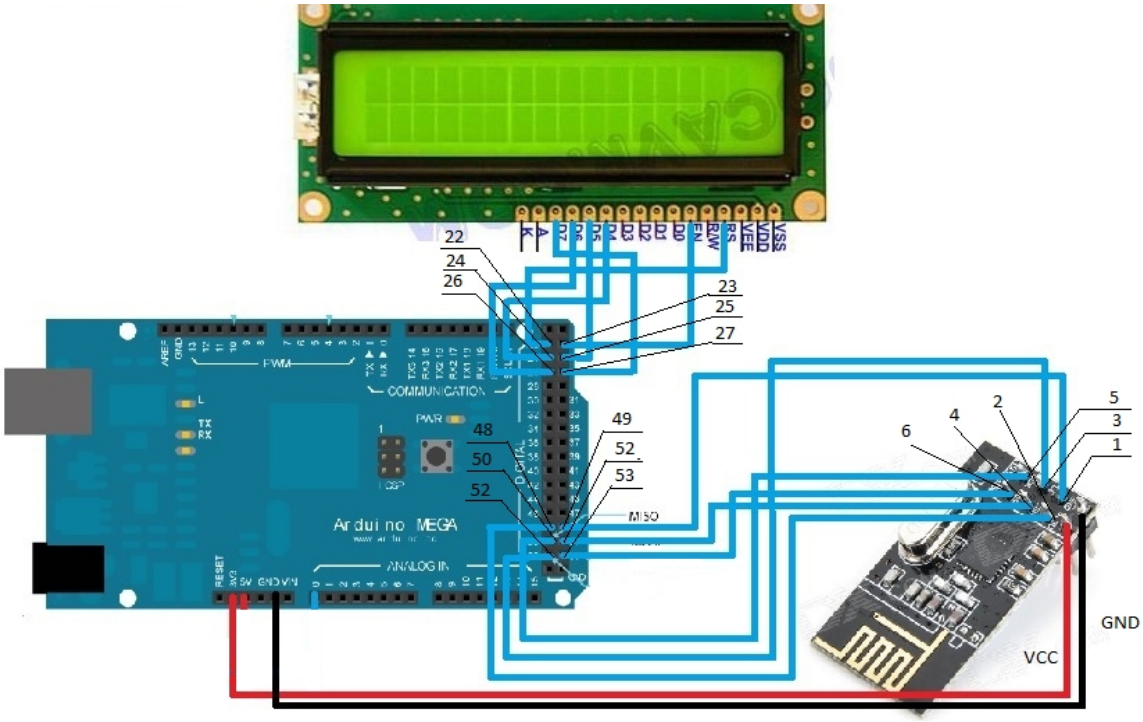
a. Bảng 4. Bảng kết nối chân Arduino với LCD:

Arduino pin	22	23	24	25	26	27
LCD pin	RS	E	D4	D5	D6	D7

b. Bảng 5. Bảng kết nối chân Arduino với module NRF24L01:

Arduino pin	GND	3.3V	8	7	52	51	50	53
NRF24L01 PIN	GND	VCC	CE	CSN	SCK	MOSI	MISO	IRQ

a. Sơ đồ kết nối chung cho các khối:



Hình 2.1: Sơ đồ kết nối phần cứng bên thu.

4. Lập trình cho arduino mạch thu:

4.1. Giới thiệu thư viện sử dụng trong Arduino bên thu:

+ Thư viện: module nRF24L01

```
#include <nRF24L01p.h>
```

+Thư viện: truyền dữ liệu

```
#include <SPI.h>
```

+ Thư viện: LCD

```
#include <LiquidCrystal.h>
```

4.2 CODE chương trình Arduino:

```
//khai bao thu vien LCD
```

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <nRF24L01p.h>
LiquidCrystal lcd(22, 23, 24, 25, 26, 27);//LiquidCrystal(rs, enable, d4,
d5, d6, d7)//dieu khien LCD qua 6 chan cua LCD
nRF24L01p receiver(48,49);//CSN,CE
int index,index_0;
int Temp,ND_Temp;
float T;

//ham xuat ra ki tu do C
byte ki_tu_do[8] = {
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
};

//ham tryen du lieu den may tinh
void serial_monitor()
{
Serial.print("Nhiet do nhan duoc la: ");//truyen du lieu khong co ki tu
xuong dong
Serial.print(T);//ghi chuoi
Serial.println(" do C");//truyen du lieu co ki tu xuong dong
```

```
}

//ham nhan du lieu
void nhan()
{
  receiver.read();
  receiver.rxPL(Temp);
}

void setup(){
  lcd.createChar(0, ki_tu_do);//tao ki tu do "0"  ki tu do co ten "char(0)"
  //khoi tao LCD
  lcd.begin(16,2);//hien thi LCD 16 cot 2 hang
  //hien thi hang 1
  lcd.setCursor(0, 0);//lcd.setCursor(col, row) di chuyen con tro tai cot 0
hang 0
  lcd.print("Chanel 1:");

  //xuat ki tu char(0) ra LCD
  lcd.setCursor ( 14, 0 );
  lcd.print(char(0));

  //xuat ki tu "C" ra LCD
  lcd.setCursor(15,0);
  lcd.print("C");
  //hien thi hang 2
  lcd.setCursor(0, 1);//lcd.setCursor(col, row) di chuyen con tro tai cot 0
hang 0
  lcd.print("Chanel 2:");
```

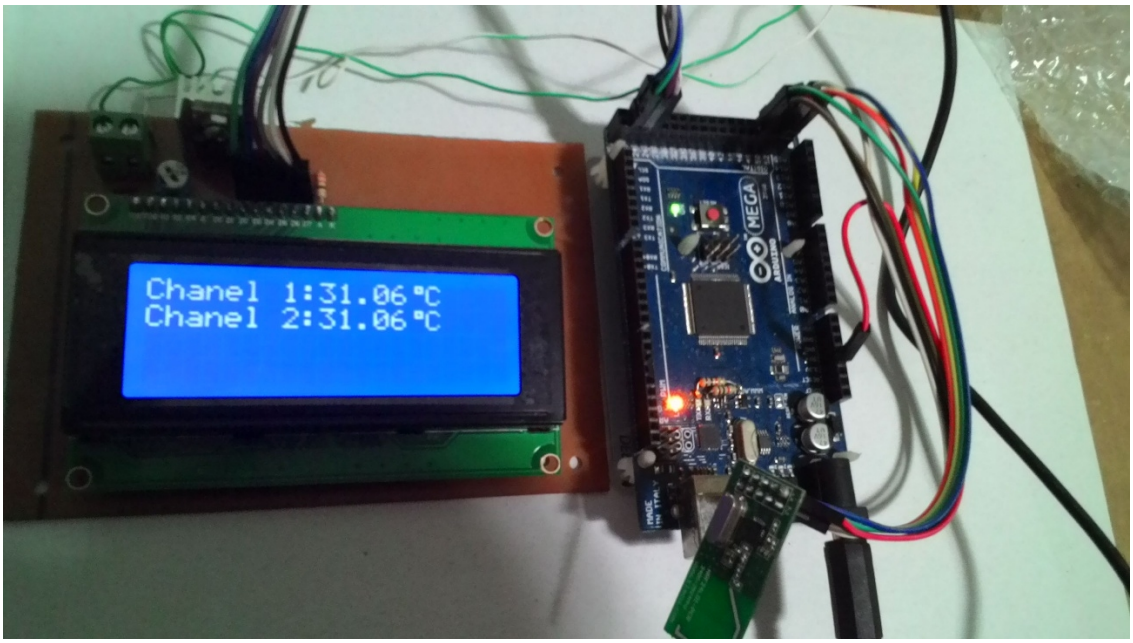
```
//xuat ki tu char(0) ra LCD
lcd.setCursor ( 14, 1 );
lcd.print(char(0));

//xuat ki tu "C" ra LCD
lcd.setCursor(15,1);
lcd.print("C");

//khoi tao toc do truyen du lieu noi tiep den may tinh
Serial.begin(9600);
Serial.println("Nhan nhiet do.....");
//khoi tao truyen du lieu SPI
SPI.begin();
//khoi tao nRF24L01
receiver.channel(10);
receiver.RXaddress("Artur");
receiver.init();
index_0=0;
//dong bo du lieu nhan va phat
while(!receiver.available());
do
{
   if(receiver.available())
   {
     receiver.read();
     receiver.rxPL(Temp);
     //Serial.println(Temp);
     if(Temp==0 || Temp==1)
```

```
        {
          index = Temp;
          Serial.println(index);
        }
       }
  }
  while(index==index_0);

}
void loop(){
  //xuat du lieu nhiet do ra LCD
    while(!receiver.available());
     if(receiver.available())
       {
       receiver.read();
       receiver.rxPL(ND_Temp);
       T=ND_Temp;
       T=T/100;
       lcd.setCursor(9,index);
       //Serial.println(ND_Temp);
       Serial.println(T);
       lcd.print(T);

  while(!receiver.available());
  if(receiver.available())
  {
    receiver.read();
    receiver.rxPL(Temp);
    if(Temp== 0 || Temp==1)
```

```
        {
           index = Temp;
          Serial.println(index);


        }


      }
         }
    }
```

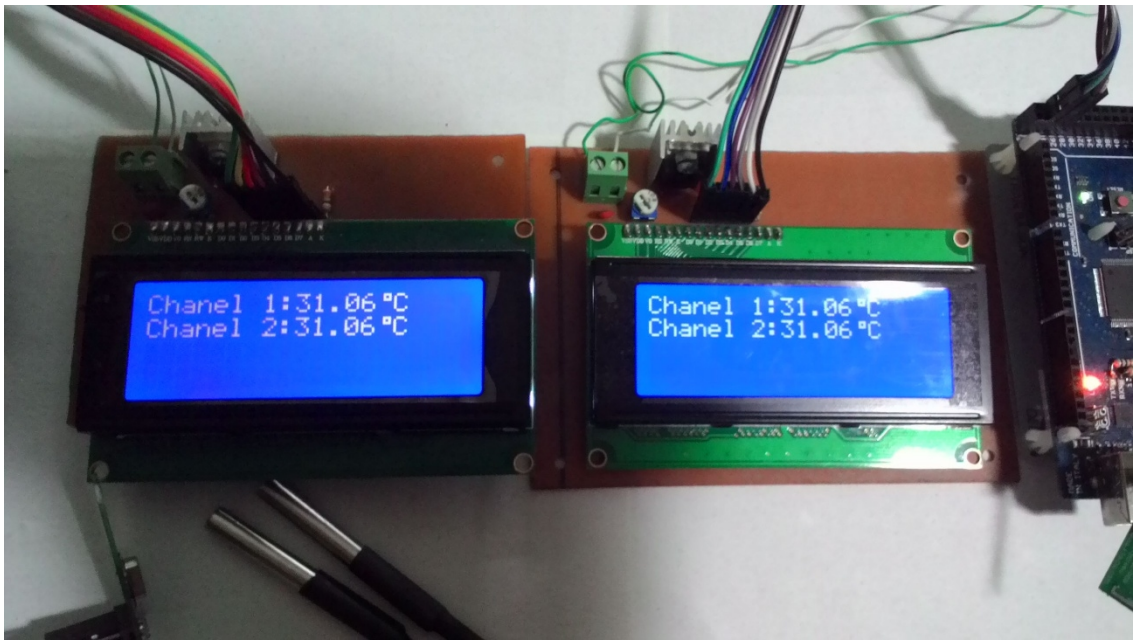
5. Nạp code và chạy chương trình:

– Sau khi nạp chương trinh cho Ardiuno, thì dữ liệu nhiệt độ sẽ thu thông qua module NRF24L01 và hiển thị lên LCD, ta cũng có thể hiển thị giá trị nhiệt độ này lên màn hình máy tính thông qua chế độ Serial Monitor của phần mềm Arduino IDE.



Hình 2.2: Hình ảnh thực tế kết quả nhiệt độ nhận được bên thu.

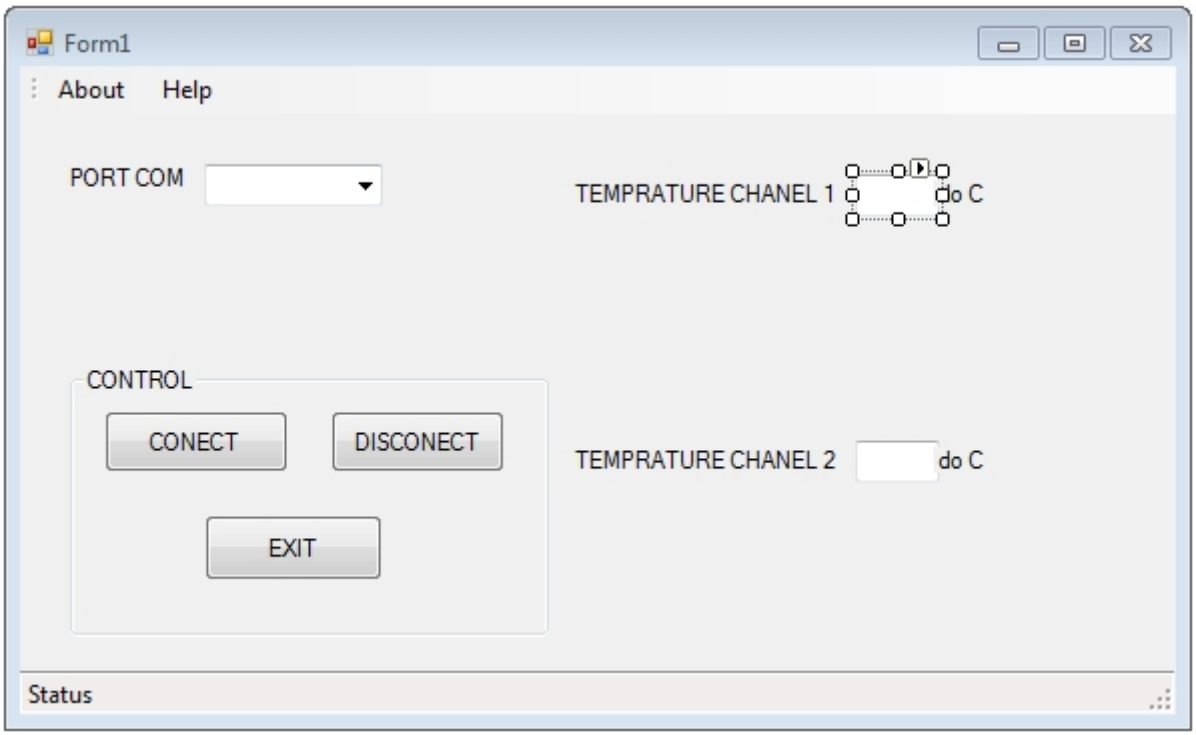
6. Kiểm tra sự đồng bộ giữa bên phát và bên thu:



Hình 2.3: Hình ảnh nhiệt độ bên phát và bên thu.

III. GIAO DIỆN NHẬN VÀ HIỂN THỊ NHIỆT ĐỘ

1. Thiết kế giao diện hiển thị nhiệt độ:



Hình 2.4: Giao diện hiển thị nhiệt độ nhận.

2. Viết chương trình cho giao diện:

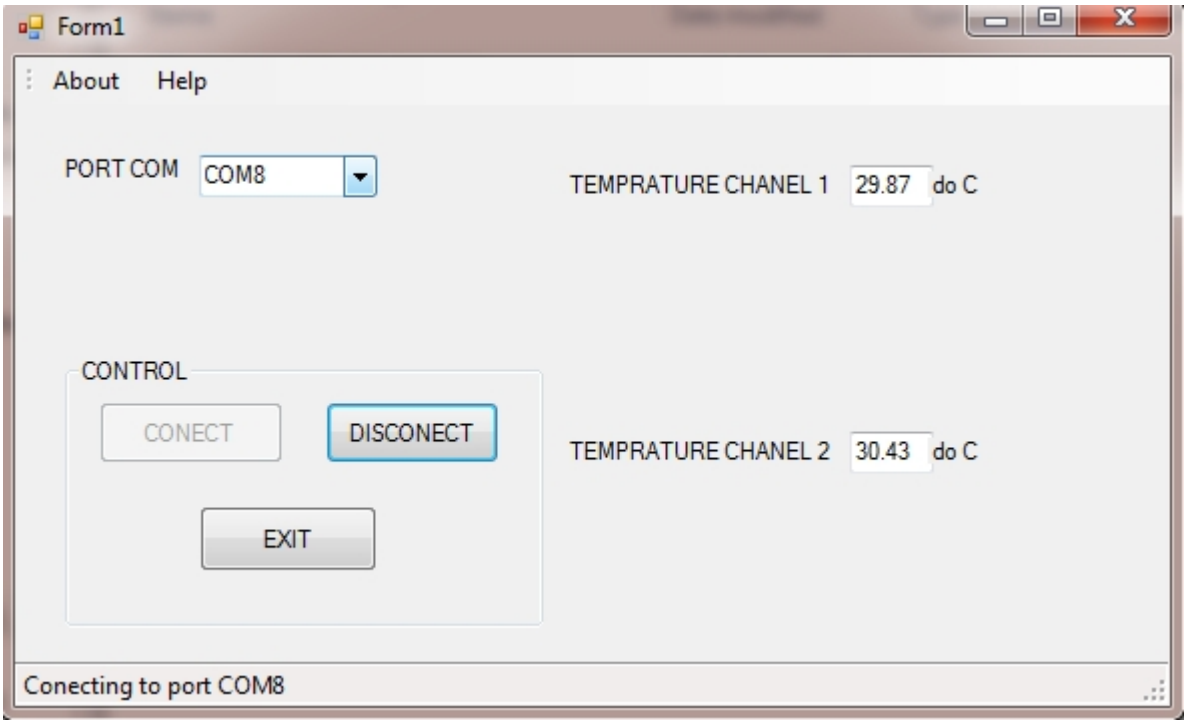
```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO;
using System.IO.Ports;
using System.Xml;

namespace RECIEVER_DATA
{
    public partial class Form1 : Form
    {
        SerialPort P = new SerialPort();
        string InputData = String.Empty;
        int TT = 0;
        delegate void SetTextCallback(string text);

        public Form1()
        {
            InitializeComponent();
            string[] ports = SerialPort.GetPortNames();
            cbCom.Items.AddRange(ports);            P.ReadTimeout = 1000;
            P.DataReceived += new SerialDataReceivedEventHandler(DataReceive);

        }
        private void DataReceive(object obj, SerialDataReceivedEventArgs e)
        {
            InputData = P.ReadLine();
            if (InputData != String.Empty)
            {

                if (Convert.ToDecimal(InputData) == 0)
                {
                    TT = 0;

                }
                else if (Convert.ToDecimal(InputData) == 1)
                {
                    TT = 1;

                }
                else
                {

                    if (TT == 0)
                    {
                        SetText1(InputData);
                    }
                    else if (TT == 1)
                    {
                        SetText2(InputData);
                    }

                }

            }

        }
```

```
        // Hàm hien thi nhiet do kenh 1
        private void SetText1(string text1)
        {
            if (this.textBox1.InvokeRequired)
            {
            SetTextCallback d = new SetTextCallback(SetText1);
this.Invoke(d, new object[] { text1 });
            }
            else this.textBox1.Text = text1;
        }

        // Hàm hien thi nhiet do kenh 2
        private void SetText2(string text2)
        {
            if (this.textBox2.InvokeRequired)
            {
                SetTextCallback d = new SetTextCallback(SetText2);
this.Invoke(d, new object[] { text2 });
            }
            else this.textBox2.Text = text2;
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            cbCom.SelectedIndex = 0;
            P.BaudRate = 9600;
            P.DataBits = 8;
            P.Parity = Parity.None;
            P.StopBits = StopBits.One;
        }


        private void btKetNoi_Click_1(object sender, EventArgs e)
        {
            try
            {
                P.Open();
                btNgat.Enabled = true;
                btKetNoi.Enabled = false;
                // Hiện thị Status
                StripStatus.Text = "Conecting to port " +
cbCom.SelectedItem.ToString();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Can't conect.", "Try again", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void btNgat_Click_1(object sender, EventArgs e)
        {
            P.Close();
            btKetNoi.Enabled = true;
            btNgat.Enabled = false;
            // Hiện thị Status
            StripStatus.Text = "Disconected !";
        }

        private void btThoat_Click_1(object sender, EventArgs e)
        {
            DialogResult kq = MessageBox.Show("Do you want exit ?", "Thong bao",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
```

```
            if (kq == DialogResult.Yes)
            {
                this.Close();
            }
        }

        private void cbCom_SelectedIndexChanged_1(object sender, EventArgs e)
        {
            if (P.IsOpen)
            {
                P.Close();
            }
            P.PortName = cbCom.SelectedItem.ToString();        }

        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void toolStripMenuItem1_Click(object sender, EventArgs e)
        {
            Form2 frmForm2 = new Form2();
            frmForm2.Show();
        }

        private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form3 frmForm3 = new Form3();
            frmForm3.Show();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }



    }
}
```

6.3 Biên dịch cài đặt file setup của giao diện,giao tiếp với Arduino



Hình 2.5: Nhiệt độ nhận được sau khi giao tiếp với Arduino.

PHẦN 3
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

I. KẾT LUẬN:

1. Những điều đề tài làm được:

Như vậy, với đề tài đồ án 1:**ĐO NHIỆT ĐỘ DÙNG CẢM BIẾN DS18B20 SỬ DỤNG BOARD ARDUINO,HIỂN THỊ TRÊN LCD,TRUYỀN PHÁT KHÔNG DÂY,GIAO TIẾP VỚI MÁY TÍNH QUA CỔNG COM.**

Đề tài đã đạt được những vấn đề sau:

- Đã giới thiệu sơ lược về các phần cứng một cách chi tiết dễ hiểu
- Đã giới thiệu phần mền cần thiết thuận tiện cho việc lập trình, đồng thời chỉ ra những câu lệnh và hàm quan trong liên quan.
- Có thể đo và hiển thị nhiệt độ tương đối chính xác,tiết kiệm được số chân VDK do sử dụng cảm biến one wire.
- Có thể truyền phát giữa thiết bị thu và nhận.
- Đo nhiệt độ thang Celsius (°C).
- Tạo được giao diện nhận nhiệt độ chạy trên Windown.
- Hiển thị được nhiệt độ nhận được trên giao diện Windown đã xây dựng.

2. Những điều khó khăn gặp phải khi làm đề tài:

Đề tài đồ án 1:ĐO NHIỆT ĐỘ DÙNG CẢM BIẾN DS18B20 SỬ DỤNG BOARD ARDUINO,HIỂN THỊ TRÊN LCD,TRUYỀN PHÁT KHÔNG DÂY,GIAO TIẾP VỚI MÁY TÍNH QUA CỔNG COM. Trong quá trình thực hiện, lập trình cho mạch đo nhiệt độ, đã gặp phải nhiều khó khăn khác nhau như: do phải nghiên cứu nhiều tài liệu nước ngoài, datasheets,... dẫn đến nhiều chỗ dịch sai, dịch nhầm dẫn đến áp dụng các hàm, câu lệnh bị sai ý nghĩa, cấu trúc..., trong quá trình viết code gặp phải nhiều lỗi phát sinh mà không tìm ngay ra nguyên nhân cần đầu tư thời gian để giải quyết, nhiều linh kiện rất khó để tìm được thư viện chuẩn để lập trình... Quá trình lắp mạch cũng gặp phải những khó khăn nhất

định tuy nhiên em đã cố gắng giải quyết được vấn đề phát sinh để hoàn thành được đề tài. Em đã hoàn thành thiết kế, lập trình và lắp đặt mạch đo nhiệt độ sử dụng Arduino không truyền phát trong vòng 2 tuần kể từ khi nhận đề tài và sau đó nghiên cứu trong 3 tuần tiếp theo để thực hiện được chức năng truyền phát tín hiệu nhiệt độ giữa 2 board Arduino,trong 3 tuần tiếp theo em đã xây dựng được giao diện nhận nhiệt độ,2 tuần kế tiếp em đã hoàn thành giao diện nhận nhiệt độ và hiển thị được nhiệt độ nhận lên giao diện.

Do đây mới là lần đầu tiên làm một đề tài đồ án, cộng với kiến thức còn nhiều hạn chế, em tự thấy đề tài của mình thực hiện được vẫn còn rất nhiều sai sót, khiếm khuyết. Em rất mong được sự ủng hộ và giúp đỡ của thầy giáo để đề tài của em thực hiện được hoàn thiện hơn và có thêm nhiều cải tiến đáng kể và ứng dụng tốt hơn vào thực tiễn.

3. HƯỚNG PHÁT TRIỂN:

- Xây dựng hệ thống giám và điều khiển thiết bị không dây thông qua module NRF24L01 kết hợp với giao diện xây dựng trên Windown.

TÀI LIỆU THAM KHẢO

[1] Massimo Banzi, *Getting Started with Arduino*, O'Reilly Media, Inc, 2009.

[2] Michael Margollis and Nicholas Weldin, *Arduino Cookbook*, O'Reilly Media, Inc, 2011.

[3] http://arduino.cc/, truy nhập cuối cùng ngày 27/11/2013.

[4] http://arduino4projects.com/, truy nhập cuối cùng ngày 6/10/2013.

[5] http://randomnerdtutorials.com/., truy nhập cuối cùng ngày 6/10/2013.

[6] http://techshowvn.com/, truy nhập cuối cùng ngày 6/10/2013.

[7] http://www.airspavce.com/mikem/arduino/RF22/, truy nhập cuối cùng ngày 27/11/2013

[8] http://groups.google.com/group/rf22-arduino/, truy nhập cuối cùng ngày 26/11/2013

[9] http://electrodragon.com/. truy nhập cuối cùng ngày 19/11//2013.

[10] http://blogembarcado.blogspot.de/, truy nhập cuối cùng ngày 29/11/2013.

[11] http://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo/, truy nhập cuối cùng ngày 20/12/2013.

[12] http://www.voutube.com/channel/UCGSloFkUnaUknE-Z21 gmmvw?feature=watchA truy nhập cuối cùng ngày 20/12/2013.

[13] http://www.mediafire.com/download/v6bnQa7g3ep3y7o/nRP24L01p.rar/., truy nhập cuối cùng ngày 20/12/2013.