

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT**



BÁO CÁO HỌC THUẬT

**NGHIÊN CỨU LẬP TRÌNH ARDUINO
SỬ DỤNG ARDUINO IDE VÀ PROTEUS CHO MÔN HỌC
“KIẾN TRÚC VÀ HẠ TẦNG MẠNG IOT”**

Cán bộ thực hiện: Th.S. Phạm Đình Tân

Đơn vị: Bộ môn Mạng máy tính

Khoa Công nghệ thông tin

Hà Nội - 06/2020

MỤC LỤC

MỤC LỤC.....	1
CHỦ ĐỀ 1 : GIỚI THIỆU VỀ ARDUINO.....	2
1.1 Giới thiệu về Arduino	2
1.2 Bo mạch Arduino UNO	3
1.3 Lập trình Arduino	6
CHỦ ĐỀ 2 : HƯỚNG DẪN CÀI ĐẶT PROTEUS.....	8
CHỦ ĐỀ 3 : LẬP TRÌNH ỨNG DỤNG ARDUINO	11
3.1 Mạch báo cháy	11
3.2 Mạch phát hiện chuyển động.....	12
3.3 Mạch hiển thị LCD	12
3.4 Mạch thu tọa độ GPS	13
TÀI LIỆU THAM KHẢO.....	14

CHỦ ĐỀ 1: GIỚI THIỆU VỀ ARDUINO

1.1 Giới thiệu về Arduino

Arduino là một nền tảng tạo mẫu và hệ sinh thái điện tử mã nguồn mở. Nó dựa trên môi trường phần cứng và phần mềm dễ sử dụng. Nó dành cho sinh viên, nghệ sĩ, nhà thiết kế, người có sở thích, người đam mê và bất cứ ai quan tâm đến việc tạo ra các đối tượng hoặc môi trường tương tác.

Trước khi bắt đầu với Arduino, chúng ta cần hiểu vi điều khiển là gì. Điều này là do, về cơ bản, Arduino là một nền tảng vi điều khiển. Vi điều khiển là một máy tính nhỏ trên một Mạch tích hợp (IC). Nó là một gói hoàn chỉnh với một Bộ vi xử lý, bộ nhớ trên bo mạch và các thiết bị ngoại vi Nhập / Xuất có thể lập trình. Vi điều khiển được sử dụng nhiều trong các ứng dụng nhúng.

AVR là một họ vi điều khiển được phát triển bởi Tập đoàn Atmel. Một quả dưa chuột là nước Mỹ thiết kế dựa trên và nhà sản xuất của vi điều khiển. Atmel bắt đầu phát triển Bộ vi điều khiển AVR vào đầu năm 1996. Một bộ vi điều khiển VR được sửa đổi Kiến trúc Harvard chip đơn 8 bit RISC (Máy tính với tập lệnh rút gọn) vi điều khiển. Một tính năng đặc biệt của gia đình AVR là nó là một trong những họ vi điều khiển đầu tiên có bộ nhớ flash chip. Các họ vi điều khiển cạnh tranh khác tại thời điểm đó (cuối thập niên 90) có ROM, EPROM hoặc EEPROM cho chương trình và chương trình cơ sở lưu trữ.

Sức mạnh thực sự của nền tảng Arduino nằm ở chỗ nó dành cho tất cả mọi người. Arduino ban đầu có nghĩa là cho các sinh viên. Mục đích của nó là cung cấp một nền tảng và hệ sinh thái nguồn mở và thấp cho sinh viên học điện tử và lập trình. Thời gian trôi qua, sự phổ biến của Arduino ngày càng tăng và nó tràn ngập trong nhiều khu vực. Ngày nay, Arduino được sử dụng nổi bật như là một nền tảng vi điều khiển ưa thích nhất trong các tổ chức giáo dục và học thuật. Nó cũng được sử dụng rộng rãi trong các hệ thống nhúng trong các lĩnh vực công nghiệp, y tế, khai thác mỏ và giám sát giao thông. Nó cũng đã tìm thấy vị trí trong nghiên cứu tích cực trong các lĩnh vực mô hình hóa, mô phỏng và giao diện người-máy tính.

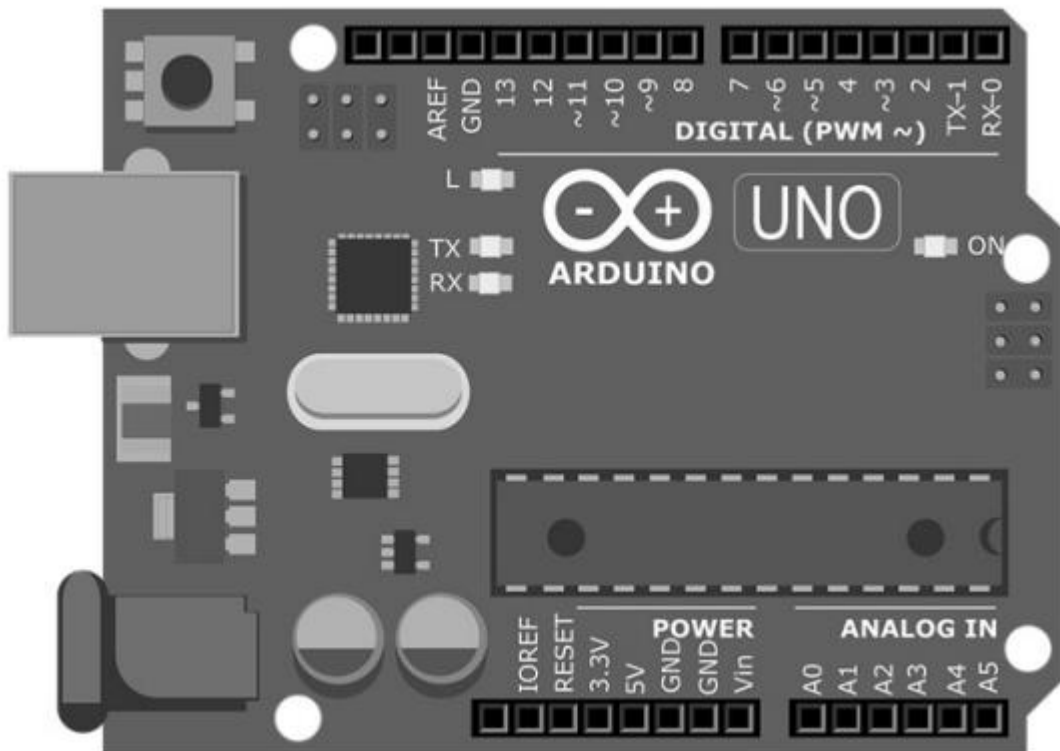
1.2 Bo mạch Arduino UNO

Arduino Uno là bảng tốt nhất cho những người mới bắt đầu với Arduino nền tảng lần đầu tiên. Nó là bảng tài liệu nhất và được sử dụng rộng rãi. Nó sử dụng AT vi điều khiển mega328P. Sau đây là hình ảnh của Arduino UNO REV 3,



Hình 1: Bo mạch Arduino Uno

Arduino Uno dựa trên vi điều khiển ATmega328P. Nó có bộ nhớ flash 32 KB, trong đó 0,5 KB được sử dụng bởi bộ tải khởi động. Bộ tải khởi động là một chương trình nhỏ chạy mọi lúc khi vi điều khiển được cấp nguồn hoặc đặt lại. Nó về cơ bản cho vi điều khiển biết phải làm gì tiếp theo khi được bật nguồn. Nó là loại Hệ điều hành dùng cho vi điều khiển. Bộ tải khởi động được tải sẵn trên bộ nhớ flash của vi điều khiển ATmega328P được cài đặt trên Arduino Uno.



Hình 2: Sơ đồ chân của bo mạch Arduino Uno Rev3

Chân dưới cùng bên trái không được sử dụng.

Chân **IOREF** là để cung cấp điện áp tham chiếu logic. Nó được kết nối với nguồn 5V

Chân **RESET** được sử dụng để thiết lập lại vi điều khiển bằng cách đưa nó xuống thấp. Chúng ta hãy nhìn vào các chân nguồn.

Chân **3.3V** cung cấp công suất quy định là 3,3V. Dòng cho phép tối đa là 50 mA.

Chân **5V** cung cấp điện áp 5V.

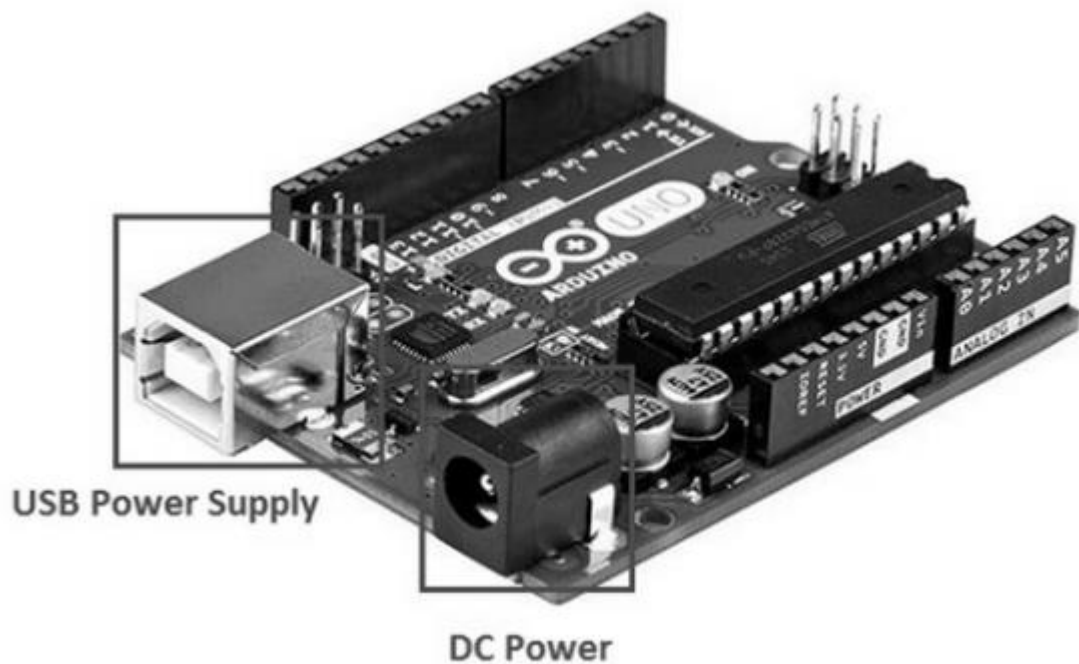
Các chân **GND** là các chân nối đất.

Chân **Vin** hoạt động như một chân nhận cấp nguồn 9-12V từ bên ngoài cho mạch. Nếu bo mạch được cấp nguồn bởi giắc cắm nguồn hoặc USB, chân này sẽ cho ra điện áp 5V.

Hãy cùng xem các chân đầu vào Analog của Arduino Uno. Những cái này nằm ở phía dưới góc phải trong hình trên. Có sáu chân tương tự A0 đến A5 và chúng được sử dụng để đọc các tín hiệu từ các cảm biến tương tự. Mỗi chân này có

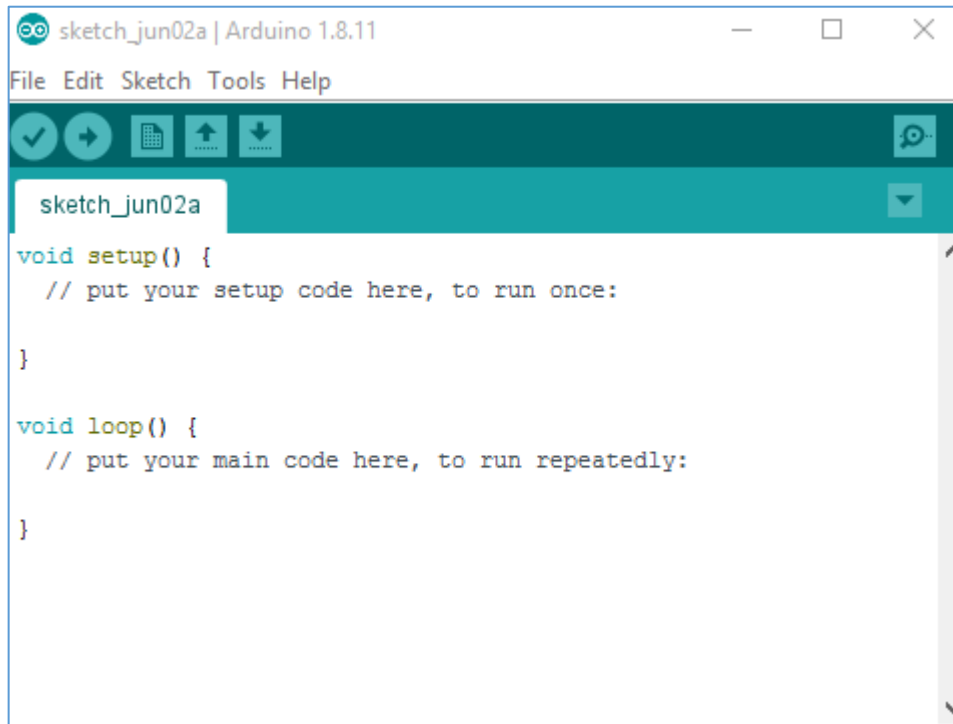
độ phân giải 10 bit. Nó có thể hoạt động với 1024 (2^{10}) giá trị khác nhau (mức điện áp).

Có mười bốn chân vào/ra số trên bo mạch Arduino Uno. Chúng có thể được sử dụng làm đầu vào và đầu ra số tùy theo chế độ. Chúng được đánh số 0 đến 13. Chúng hoạt động ở mức 5V và có thể cung cấp hoặc nhận dòng điện 20mA. Nếu dòng điện nhận/cấp vượt quá 40mA ở các chân này có thể làm hỏng bo mạch. Ngoài chức năng vào/ra số, các chân 3, 5, 6, 9, 10 và 11 còn được sử dụng cho 8 bit đầu ra PWM (Điều chế độ rộng xung). Trong tất cả các chân vào/ra số, một số chân còn có chức năng riêng biệt. Thông tin nổi tiếp: chân 0 (RX) và chân 1 (TX) được sử dụng cho giao tiếp nối tiếp. Ngắt ngoài: chân 2 và 3 được sử dụng để định cấu hình để kích hoạt bên ngoài, ngắt trên giá trị thấp, sườn xung tăng hoặc giảm hoặc thay đổi giá trị.



Hình 3: Cấp nguồn sử dụng cấp nguồn hoặc cáp USB

1.3 Lập trình Arduino



Hình 4: Giao diện Arduino IDE

Type	Byte Length	Range of values
boolean	1	true / false
char	1	-128 to +127
unsigned char	1	0 to 255
byte	1	0 to 255
int	2	-32,768 to 32,767
unsigned int	2	0 to 65,535
word	2	0 to 65,535
long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
float	4	-3.4028235E+38 to 3.4028235E+38
double	4	-3.4028235E+38 to 3.4028235E+38
string	?	A null terminated reference data type
String	?	An reference data type object
array	?	A sequence of a value type
void	0	A descriptor used with functions when they return nothing

Hình 5: Các kiểu dữ liệu lập trình trên Arduino IDE

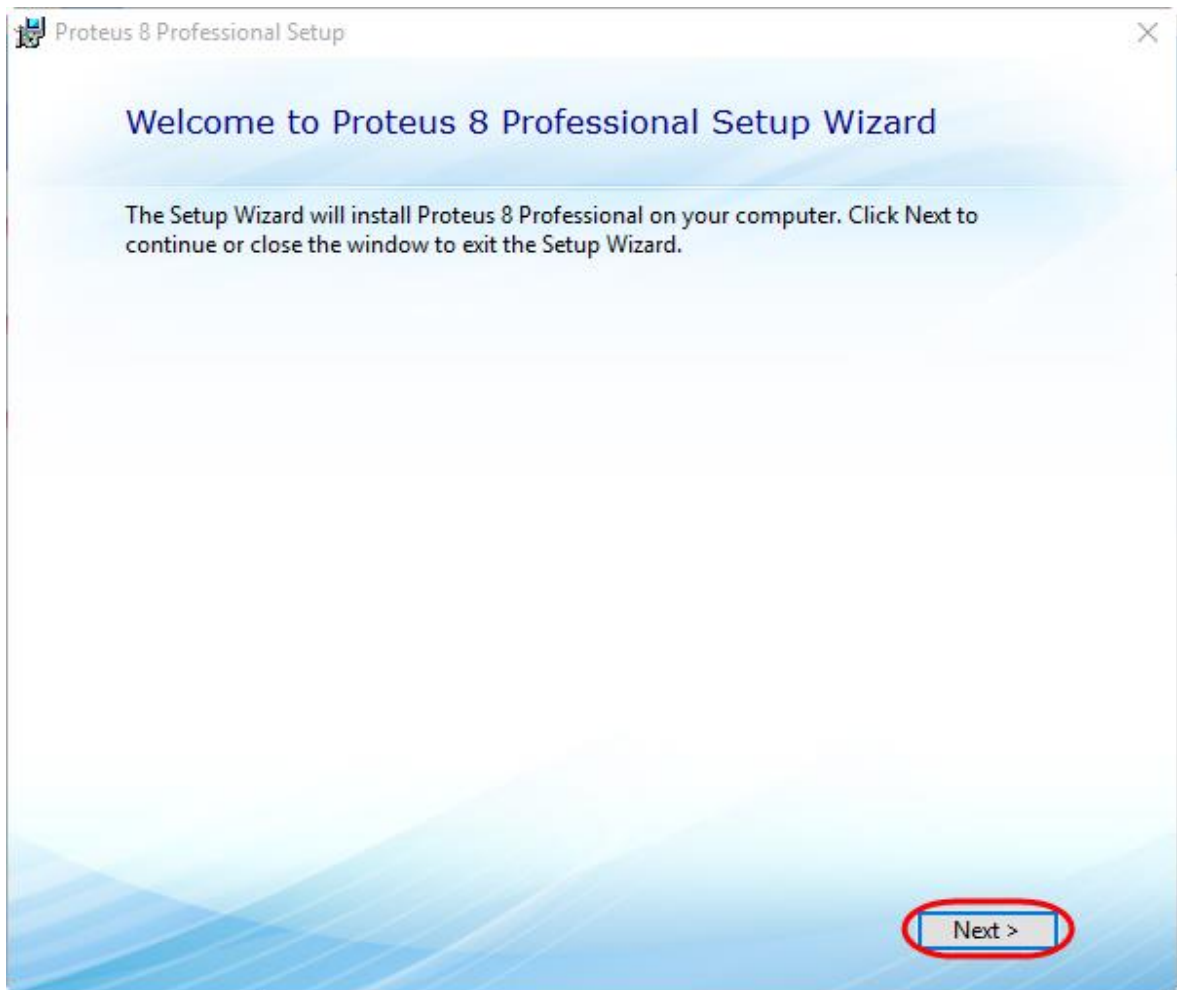
Code Construct	Description
int	Integer values, such as 123
float	Decimal values, such as 1.15
char []	String values, such as "Arduino"
HIGH	Digital pin with current
LOW	Digital pin with no current
INPUT	Pin can only be read
OUTPUT	Pin can only be set
A0 - A7	Constants for analog pins; varies by board
0 - 13	Value for digital pins; varies by board
analogRead()	Returns analog pin value (0 - 1023)
analogWrite(...)	Sets analog pin value
digitalRead()	Returns digital pin value (HIGH or LOW)

Hình 6: Các lệnh lập trình cơ bản trên Arduino IDE

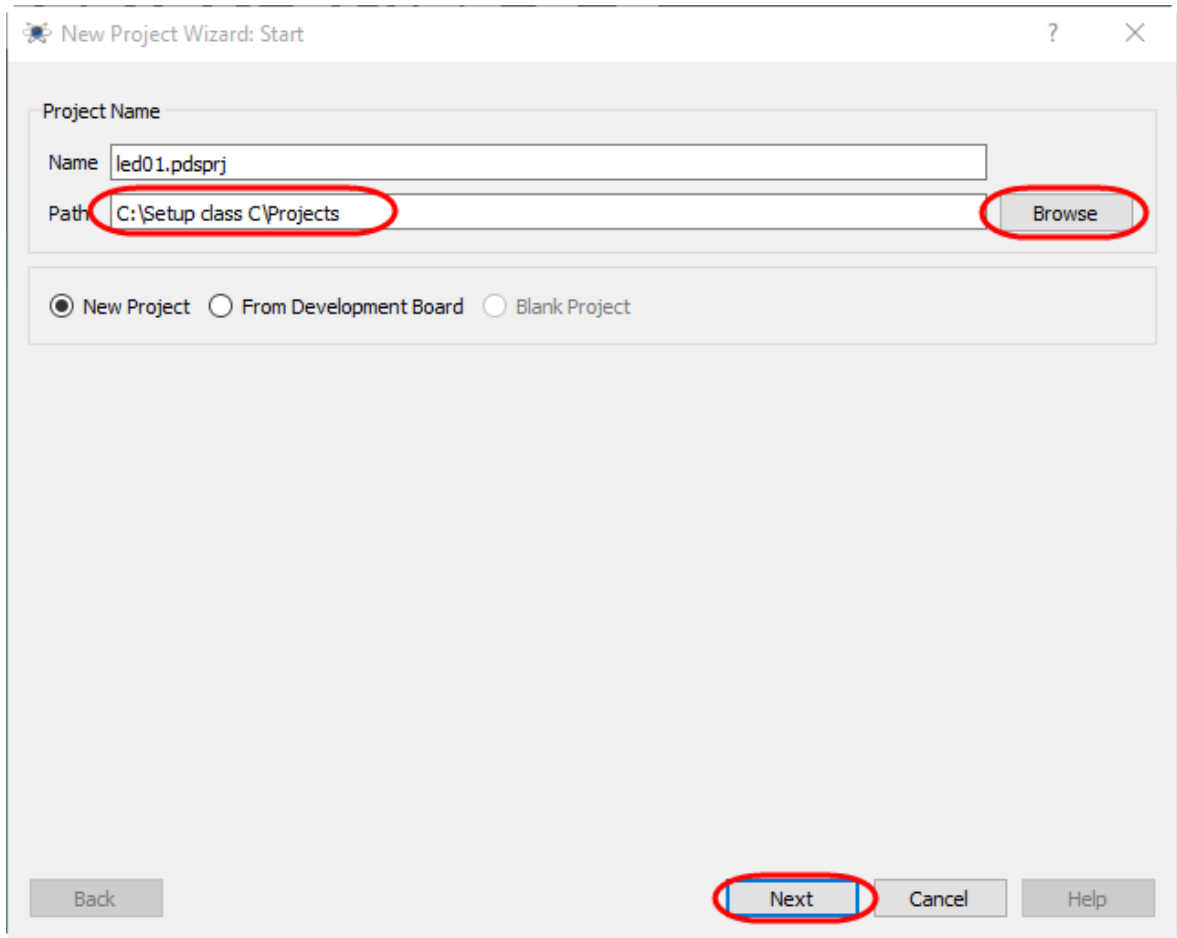
Serial.begin()	Initializes serial monitor
Serial.print()	Logs message on serial monitor
Serial.println()	Logs message on serial monitor with new line
delay(ms)	Adds a wait in processing
setup()	Standard Arduino function called once
loop()	Standard Arduino function called repeatedly
if	Checks for a true/false condition
if ... else	Checks for a true/false condition; if false goes to else
//	Single-line comment
/* */	Multiline comment
#define	Defines a constant
#include	Includes an external library

Hình 7: Các lệnh lập trình cơ bản trên Arduino IDE

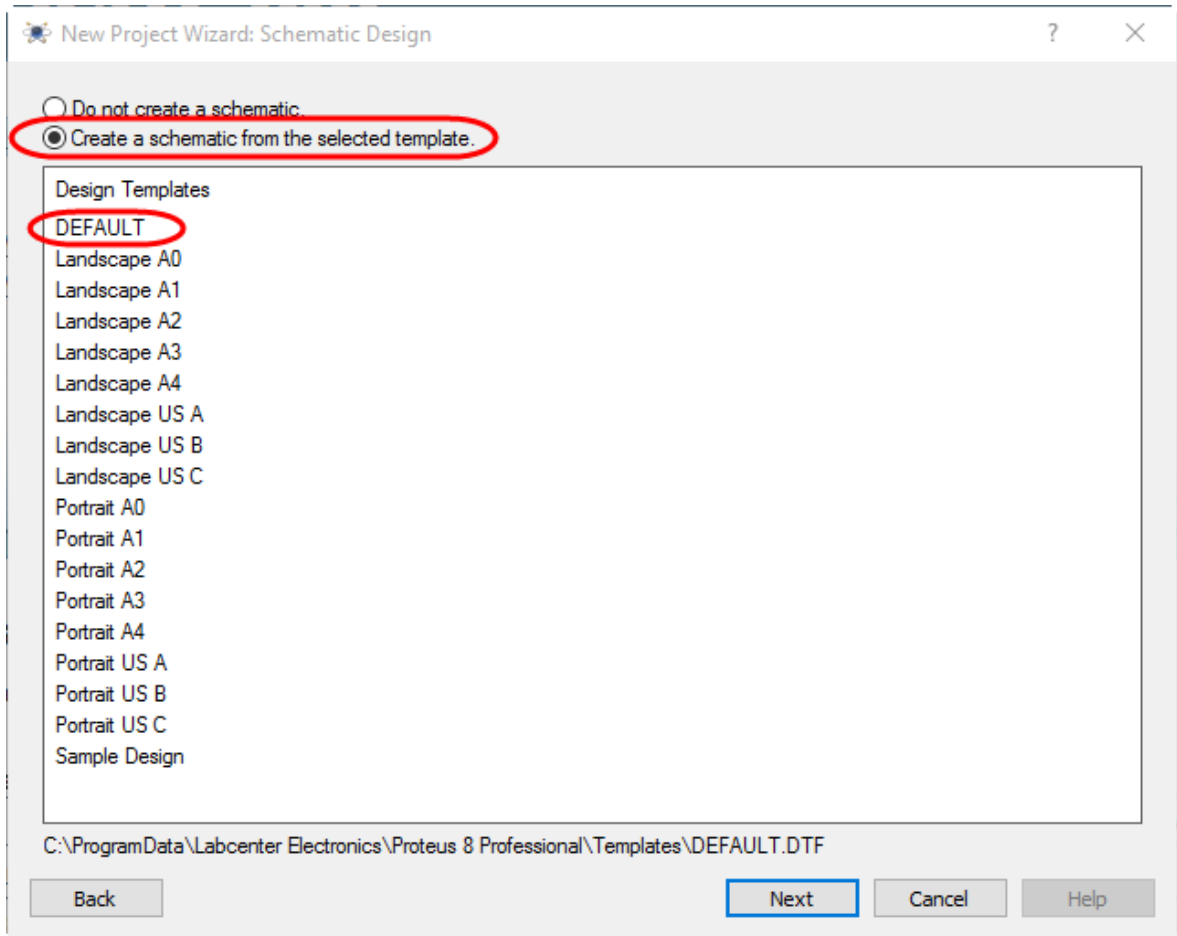
CHỦ ĐỀ 2: HƯỚNG DẪN CÀI ĐẶT PROTEUS



Hình 8: Giao diện cài đặt Proteus



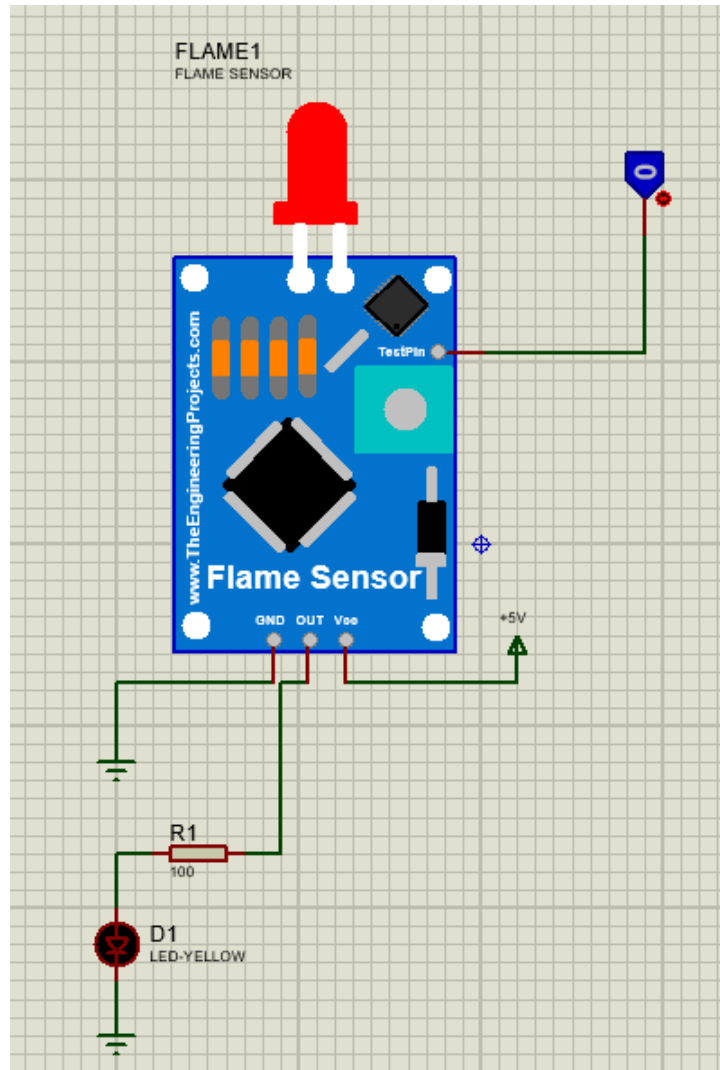
Hình 9: Tạo Project mới trên Proteus



Hình 10: Tạo Schematic mới trên Proteus

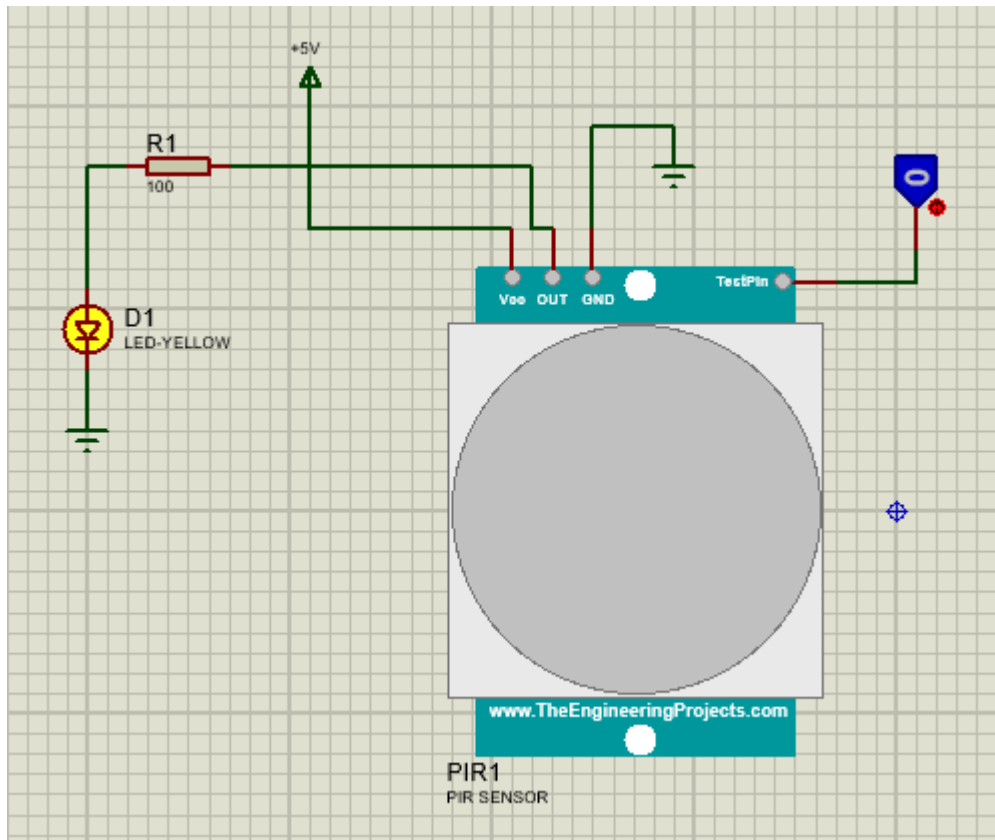
CHỦ ĐỀ 3: LẬP TRÌNH ỨNG DỤNG ARDUINO

3.1 Mạch báo cháy



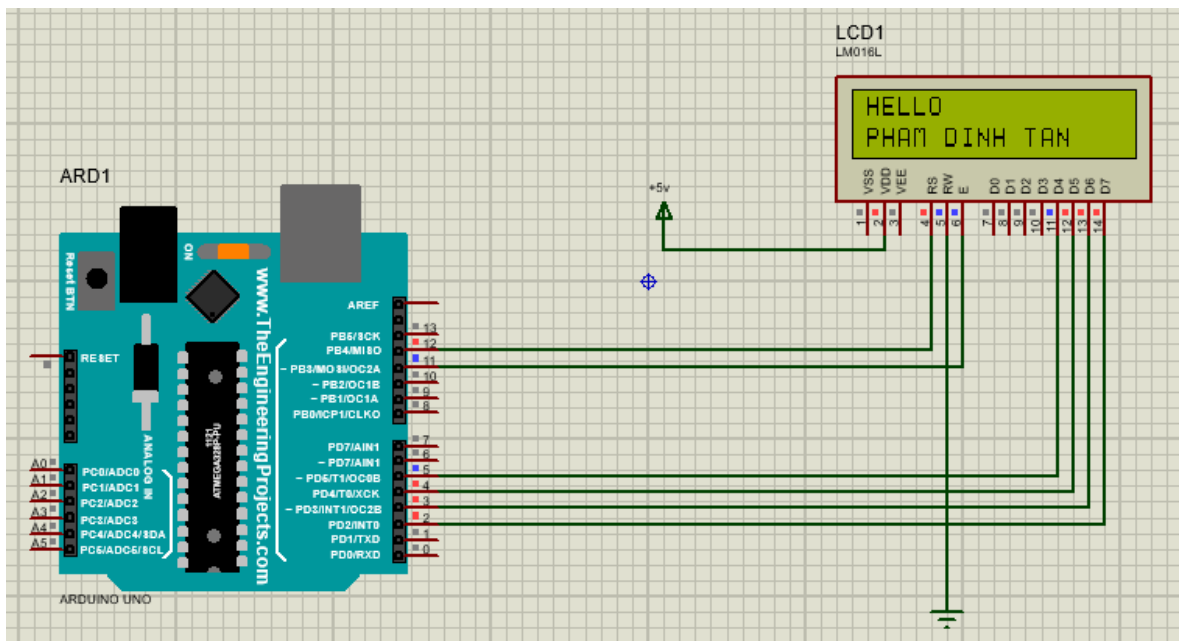
Hình 11: Sơ đồ mạch báo cháy

3.2 Mạch phát hiện chuyển động



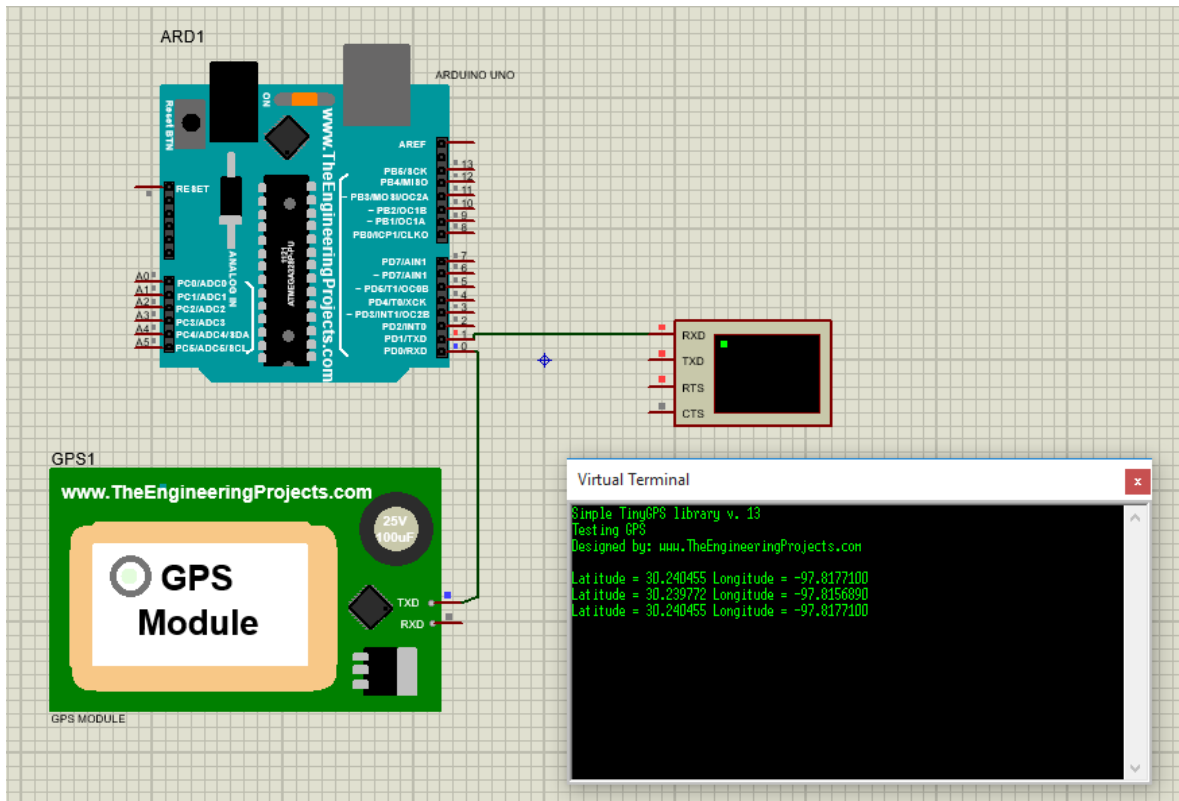
Hình 12: Sơ đồ mạch phát hiện chuyển động

3.3 Mạch hiển thị LCD



Hình 13: Sơ đồ hiển thị LCD

3.4 Mạch thu tọa độ GPS



Hình 14: Sơ đồ mạch thu tọa độ GPS

TÀI LIỆU THAM KHẢO

1. Pajankar, Ashwin. Arduino Made Simple: With Interactive Projects. BPB Publications, 2018.
2. Trang web: <https://www.theengineeringprojects.com/2016/07/flame-sensor-library-proteus.html>
3. Trang web: <https://www.theengineeringprojects.com/2016/01/pir-sensor-library-proteus.html>
4. Trang web: <https://www.theengineeringprojects.com/2014/07/circuit-designing-lcd-arduino-proteus-isis.html>
5. Trang web: <https://www.theengineeringprojects.com/2015/12/gps-library-proteus.html>
6. Trang web: <https://www.theengineeringprojects.com/2016/05/interfacing-gps-module-arduino-proteus-isis.html>
7. Trang web: <https://github.com/mikalhart/TinyGPS>