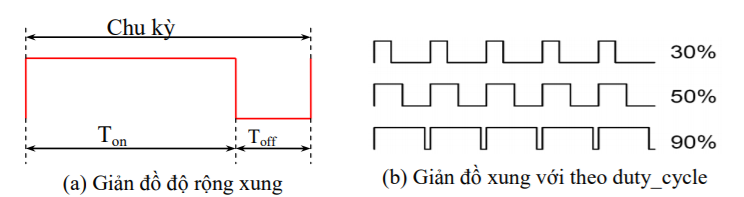
**Phần I: Điều khiển động cơ**

1. Phương pháp điều chế độ rộng xung ( PWM).

Tốc độ quay của động cơ một chiều tỉ lệ thuận với điện áp đầu vào. Do đó, cách đơn giản nhất để điều khiển tốc độ quay của rotor là thay đổi mức điện áp đặt vào động cơ (1.1). Nguyên tắc cơ bản để thay đổi tốc độ quay động cơ của phương pháp PWM là giữ nguyên giá trị điện áp vào và thay đổi thời gian đặt điện áp vào động cơ. Điều này có nghĩa, thời gian mức cao Ton trong một chu kỳ của xung ngõ vào động cơ càng lớn thì điện áp trung bình đặt vào động cơ càng cao, ngược lại thời gian mức thấp Toff trong một chu kỳ của xung ngõ vào động cơ càng lớn điện áp trung bình đặt vào động cơ càng thấp (Hình 1.1). Đại lượng mô tả mối quan hệ giữa khoảng thời gian T-on và T-off được gọi là độ rộng xung (duty\_cycle), được tính theo công thức (1.1):

Duty cycle = x 100 (1.1)



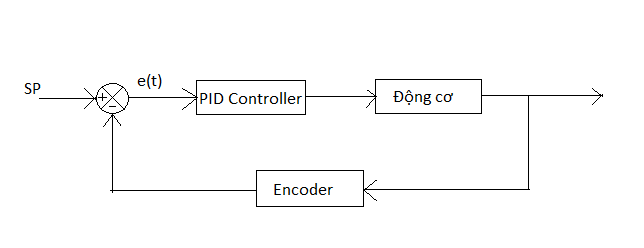
*Hình 1.1: Giản đồ thời gian của xung PWM*

Thí dụ: Trong hình 5.b, nếu chu kỳ xung 1kHz; thời gian xung mức cao Ton là 0.3ms và thời gian xung mức thấp Toff là 0.7ms. Biên độ xung là 12Vdc. Tốc độ quay của rotor là 1500 vòng/phút. Theo công thức (1.1):

duty\_cycle = 30% và điện áp trung bình đặt vào động cơ là 12 x 30% = 3.6Vdc.

Tốc độ quay của rotor là 1500 x 30% = 450 vòng/phút.

2.Sử dụng bộ điều khiển PID điều khiển tốc độ động cơ

Giải thuật điều khiển vi tích phân tỉ lệ hiệu chỉnh sai số giữa giá trị biến đo được và giá trị mong muốn đạt đến bằng cách tính toán và xuất ra một "hành động điều chỉnh" nhanh chóng để giữ cho lỗi ở mức nhỏ nhất có thể được. Bộ điều khiển này bao gồm 3 thống số: Proportional (P), Integral (I) và Derivative (D), được thể hiện qua sơ đồ khối:

*Hình 2.1: Sơ đồ khối của giải thuật PID*

Khi đặt cho động cơ một giá trị đặt SP (set point) là vset nhất định, ta sẽ tính được độ lệch giữa giá trị mong muốn và giá trị đọc được từ Encoder

e(t) = vset – v(t) (2.1)

Từ hàm lỗi e(t) đó thì bộ điều khiển PID sẽ đưa ra giá trị xung phù hợp để truyền cho động cơ.

Phương trình PID tổng quát:

U(t) = Kp\*e(t) + Ki + Kde(t) (2.2)

- Proportional P(t) (hàm tỉ lệ): điều khiển tỉ lệ với giá trị lỗi hiện tại.

P(t) = Kp.e(t) (2.3)

Trong đó: Kp: hằng số tỉ lệ

- Integral I(t) (hàm tích phân): điều khiển tương ứng với mức lỗi được tích lũy theo thời gian.

I(t) = Ki (2.4)

Trong đó: Ki : hằng số điều chỉnh hàm tích phân.

- Derivative D(t) (hàm vi phân): điều khiển tương ứng với tốc độ thay đổi của lỗi, hay độ dốc lỗi theo thời gian.

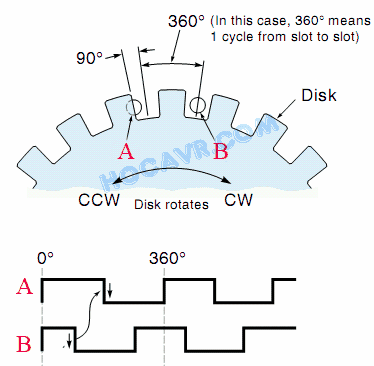
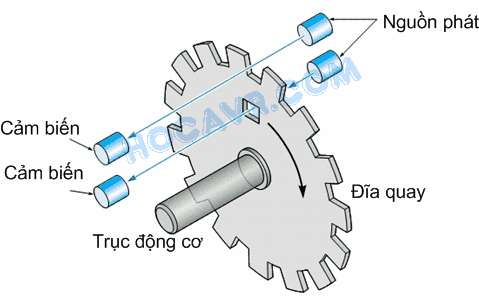
D(t) = Kde(t) (2.5)

Trong đó: Kd: hằng số điều chỉnh hàm vi phân.

Để chọn các thành phàn Kp, Ki, Kd cho bộ điều khiển PID:

 - Chọn Kp trước: thử bộ điều khiển P với đối tượng thật (hoặc mô phỏng), điều chỉnh Kp sao cho thời gian đáp ứng đủ nhanh, chấp nhận overshot nhỏ.  
     - Thêm thành phần D để loại overshot, tăng Kd từ từ, thử nghiệm và chọn giá trị thích hợp. Steady state error có thể sẽ xuất hiện.  
     - Thêm thành phần I để giảm steady state error. Nên tăng Ki từ bé đến lớn để giảm steady state error đồng thời không để cho overshot xuất hiện trở lại.

3. Đọc encoder



*Hình 3.1: Optical Encoder*

Encoder thường có 3 kênh (3 ngõ ra) bao gồm kênh A, Kênh B và Kênh I. Trong hình 3.1 có một lỗ nhỏ bên phía trong của đĩa và một cặp phát thu dành riêng cho lỗ nhỏ này. Đó là kênh I của encoder. Cứ mỗi lần motor quay được một vòng, lỗ nhỏ xuất hiện tại vị trí của cặp phát- thu, hồng ngoại từ nguồn sẽ xuyên qua lỗ nhỏ đến cảm biến quang, một tín hiệu xuất hiện trên cảm biến. Như thế kênh I xuất hiện một xung mỗi vòng quay của motor. Bênh ngoài được chia các rãnh nhỏ và một cặp phát-thu khác dành cho các rãnh này. Đây là kênh A của encoder, hoạt động của kênh A cũng tương tự kênh I, điểm khác nhau là trong 1 vòng quay của motor, có N “xung” xuất hiện trên kênh A. N là số rãnh trên đĩa và được gọi là độ phân giải (resolution) của encoder.

Trên encoder có một cặp thu-phát khác được đặt cùng đường tròn với kênh A nhưng được đặt lệch (M + 0.5) rãnh so với kênh A. Tín hiệu xung kênh B có cùng tần số so với kênh A nhưng lệch pha 90 so với kênh A. Bằng cách phối hợp kênh A và kênh B người đọc sẽ biết được chiều quay của động cơ.

Trên hình bên phải thể hiện sự bố trí của cảm biến kênh A và B lệch pha nhau. Khi motor quay cùng chiều kim đồng hồ, lúc tín hiệu A chuyển từ mức cao xuống mức thấp( cạnh xuống) thì kênh B đang ở mức thấp. Ngược lại, nếu động cơ quay ngươc chiều kim đồng hồ, lúc tín hiệu A chuyển từ mức cao xuống mức thấp thì kênh B đang ở mức cao.

**Dùng ngắt ngoài để đọc encoder:**

Có nhiều phương pháp để đọc encoder nhưng để đơn giản và chính xác thì ta sử dụng ngắt ngoài. Ta sử dụng các chân ngắt ngoài EXTI0 để đọc tín hiệu từ kênh A của encoder bánh trái, tương tự dung EXTI1 cho bánh phải. Cứ mỗi lần ngắt ngoài xảy ra, tức là có 1 xung xuất hiện ở kênh A thì trình phục vụ ngắt tự động được gọi. Trình phục vụ ngắt sẽ tăng biến đếm xung lên 1.

Ta sử dụng ngắt tràn Timer 4 để tạo 1 khoảng thời gian lấy mẫu( sampling time). Sau mỗi thời gian lấy mẫu thì ta tính lấy gia trị xung đếm được( pulse) để suy ra tốc độ động cơ.

4. Driver điều khiển L298

IC L298 là một IC tích hợp nguyên khối gồm 2 mạch cầu H bên trong. Với  
điện áp làm tăng công suất nhỏ như động cơ DC loại vừa…  
Chức năng các chân của L298  
 - 4 chân INPUT: IN1, IN2, IN3, IN4 được nối lần lượt với các chân 5, 7, 10, 12  
của L298. Đây là các chân nhận tín hiệu điều khiển.  
 - 4 chân OUTUT: OUT1, OUT2, OUT3, OUT4 (tương ứng với các chân  
INPUT) được nối với các chân 2, 3,13,14 của L298. Các chân này sẽ được nối với  
động cơ.  
 - Hai chân ENA và ENB dùng để điều khiển mạch cầu H trong L298. Nếu ở  
mức logic “1” (nối với nguồn 5V) cho phép mạch cầu H hoạt động, nếu ở mức logic  
“0” thì mạch cầu H không hoạt động  
 Điều khiển chiều quay với L298:  
 - Khi ENA = 0: Động cơ không quay với mọi đầu vào.  
20  
 - Khi ENA = 1:  
 INT1 = 1; INT2 = 0: Động cơ quay thuận.  
 INT1 = 0; INT2 = 1: Động cơ quay nghịch.  
 INT1 = INT2: Động cơ dùng ngay tức thì.  
 Với ENB cũng tương tự với INT3, INT4.  
 Mạch điều khiển động cơ DC L298 có khả năng điều khiển 2 động cơ DC, dòng tối đa 2A mỗi động cơ.

**PHẦN II: THIẾT KẾ PHẦN MỀM**

##### 1. Giới thiệu phần mềm Microsoft Visual Studio

Microsoft Visual Studio là một [môi trường phát triển tích hợp](https://vi.wikipedia.org/wiki/M%C3%B4i_tr%C6%B0%E1%BB%9Dng_ph%C3%A1t_tri%E1%BB%83n_t%C3%ADch_h%E1%BB%A3p) (IDE) từ [Microsoft.](https://vi.wikipedia.org/wiki/Microsoft) Nó được sử dụng để phát triển [chương trình máy tính](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh) cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như [Windows API,](https://vi.wikipedia.org/w/index.php?title=Windows_API&action=edit&redlink=1) [Windows Forms,](https://vi.wikipedia.org/w/index.php?title=Windows_Forms&action=edit&redlink=1) [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation) ,  [Windows Store](https://vi.wikipedia.org/wiki/Windows_Store) và [Microsoft Silverlight.](https://vi.wikipedia.org/wiki/Microsoft_Silverlight)

Nó có thể sản xuất cả hai [ngôn ngữ máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) và [mã số quản lý.](https://vi.wikipedia.org/w/index.php?title=M%C3%A3_s%E1%BB%91_qu%E1%BA%A3n_l%C3%BD&action=edit&redlink=1)

Visual Studio bao gồm một [trình soạn thảo mã](https://vi.wikipedia.org/w/index.php?title=Tr%C3%ACnh_so%E1%BA%A1n_th%E1%BA%A3o_m%C3%A3&action=edit&redlink=1) hỗ trợ [IntelliSense](https://vi.wikipedia.org/w/index.php?title=IntelliSense&action=edit&redlink=1) cũng như [cải tiến mã nguồn.](https://vi.wikipedia.org/wiki/C%E1%BA%A3i_ti%E1%BA%BFn_m%C3%A3_ngu%E1%BB%93n) Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng [giao diện ứng dụng,](https://vi.wikipedia.org/w/index.php?title=Giao_di%E1%BB%87n_%E1%BB%A9ng_d%E1%BB%A5ng&action=edit&redlink=1) [thiết kế web,](https://vi.wikipedia.org/wiki/Thi%E1%BA%BFt_k%E1%BA%BF_web) thiết kế [lớp](https://vi.wikipedia.org/wiki/L%E1%BB%9Bp_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) và thiết kế [giản đồ cơ sở dữ liệu.](https://vi.wikipedia.org/w/index.php?title=Gi%E1%BA%A3n_%C4%91%E1%BB%93_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1) Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các [hệ thống quản lý phiên bản](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_qu%E1%BA%A3n_l%C3%BD_phi%C3%AAn_b%E1%BA%A3n) (như [Subversion)](https://vi.wikipedia.org/wiki/Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các [miền ngôn ngữ cụ thể](https://vi.wikipedia.org/w/index.php?title=Mi%E1%BB%81n_ng%C3%B4n_ng%E1%BB%AF_c%E1%BB%A5_th%E1%BB%83&action=edit&redlink=1) hoặc bộ công cụ dành cho các khía cạnh khác trong [quy trình phát triển phần mềm.](https://vi.wikipedia.org/wiki/Quy_tr%C3%ACnh_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m)

Visual Studio hỗ trợ nhiều [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có [C,](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [C++/CLI](https://vi.wikipedia.org/w/index.php?title=C%2B%2B/CLI&action=edit&redlink=1) (thông qua [Visual](https://vi.wikipedia.org/wiki/Visual_C%2B%2B)

[C++)](https://vi.wikipedia.org/wiki/Visual_C%2B%2B), [VB.NET](https://vi.wikipedia.org/wiki/VB.NET) (thông qua [Visual Basic.NET)](https://vi.wikipedia.org/wiki/Visual_Basic.NET), [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng) (thông qua [Visual C#)](https://vi.wikipedia.org/w/index.php?title=Visual_C&action=edit&redlink=1) và [F#](https://vi.wikipedia.org/w/index.php?title=F_th%C4%83ng&action=edit&redlink=1) (như củaVisual Studio 201[0)](https://vi.wikipedia.org/w/index.php?title=Visual_Studio_2010&action=edit&redlink=1). Hỗ trợ cho các ngôn ngữ khác như [J++](https://vi.wikipedia.org/w/index.php?title=J%2B%2B&action=edit&redlink=1)[/J#,](https://vi.wikipedia.org/w/index.php?title=J_th%C4%83ng&action=edit&redlink=1) [Python](https://vi.wikipedia.org/wiki/Python) và [Ruby](https://vi.wikipedia.org/wiki/Ruby) thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ [XML](https://vi.wikipedia.org/wiki/XML)[/XSLT,](https://vi.wikipedia.org/wiki/XSLT) [HTML](https://vi.wikipedia.org/wiki/HTML)[/XHTML,](https://vi.wikipedia.org/wiki/XHTML) [JavaScript](https://vi.wikipedia.org/wiki/JavaScript) và [CSS.](https://vi.wikipedia.org/wiki/CSS)

#### Cấu trúc

Visual Studio không hỗ trợ cho bất kỳ ngôn ngữ lập trình nào về giải pháp hoặc công cụ thực chất, thay vào đó nó cho phép cắm chức năng được mã hóa như là một VSPackage. Khi cài đặt, các chức năng có sẵn như là một dịch vụ.

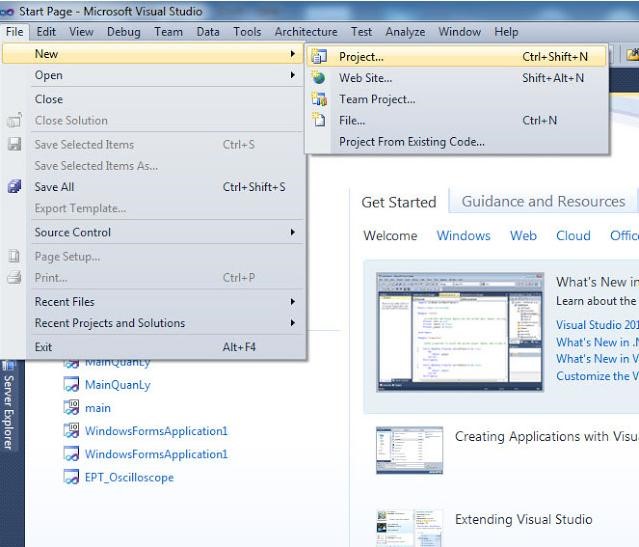
IDE cung cấp ba dịch vụ: SVsSolution cung cấp khả năng liệt kê các dự án và các giải pháp; SVsUIShell cung cấp cửa sổ và giao diện người dùng và SVsShell.

Ngoài ra, IDE cũng có trách nhiệm điều phối và cho phép truyền thông giữa các dịch vụ. Tất cả các biên tập viên, nhà thiết kế, các loại dự án và các công cụ khác được thực hiện theo VSPackages. Visual Studio sử dụng  [COM](https://vi.wikipedia.org/w/index.php?title=Component_Object_Model&action=edit&redlink=1) để truy cập VSPackages. Visual Studio [SDK](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1) cũng baogồm Managed Package Framework (MPF) là một tập hợp quản lý bao bọc quanh các COM-interfaces cho phép các gói được viết bằng bất kỳ ngôn ngữ nào. Tuy nhiên, MPF không cung cấp tất cả các chức năng bộc lộ trong Visual Studio COM-interfaces. Các dịch vụ có thể được tiêu thụ để tạo ra các gói khác, để thêm chức năng cho Visual Studio IDE.

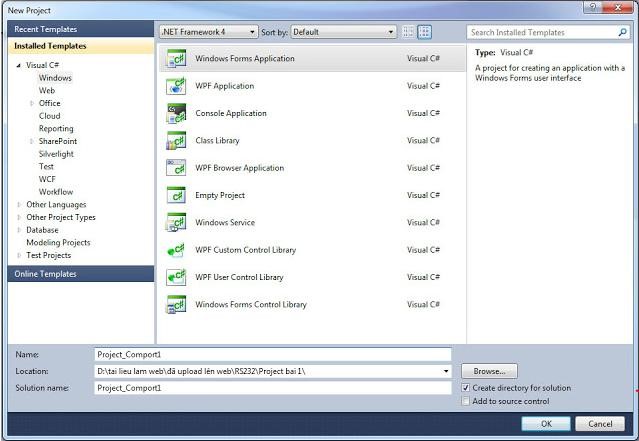
Hỗ trợ cho các ngôn ngữ lập trình được thêm vào bằng cách sử dụng một VSPackage đặc biệt được gọi là một dịch vụ ngôn ngữ. Một dịch vụ ngôn ngữ định nghĩa giao tiếp khác nhau mà việc thực hiện VSPackage có thể thực hiện để hỗ trợ thêm cho các chức năng khác nhau. Các chức năng có thể được thêm vào theo cách này bao gồm cú pháp màu, hoàn thành báo cáo kết quả, kết hợp đôi, công cụ chú giải tham số thông tin, danh sách thành viên và đánh dấu lỗi trên nền biên dịch. Nếu giao diện được thực hiện, các tính năng sẽ có sẵn ngôn ngữ. Dịch vụ ngôn ngữ sẽ được thực hiện trên cơ sở mỗi ngôn ngữ. Việc triển khai có thể tái sử dụng mã từ phân tích cú pháp hoặc trình biên dịch cho ngôn ngữ. Dịch vụ ngôn ngữ có thể được triển khai hoặc trong mã nguồn gốc hoặc mã số quản lý. Đối với mã nguồn gốc, thì cả COM-interfaces gốc hoặc Babel Framework (một phần của Visual Studio SDK) đều có thể được sử dụng. Đối với mã số quản lý thì các MPF sẽ bao hàm các dịch vu quản lý văn bản.

##### 2. Thiết kế giao diện điều khiển trên phần mềm Microsoft Visual Studio

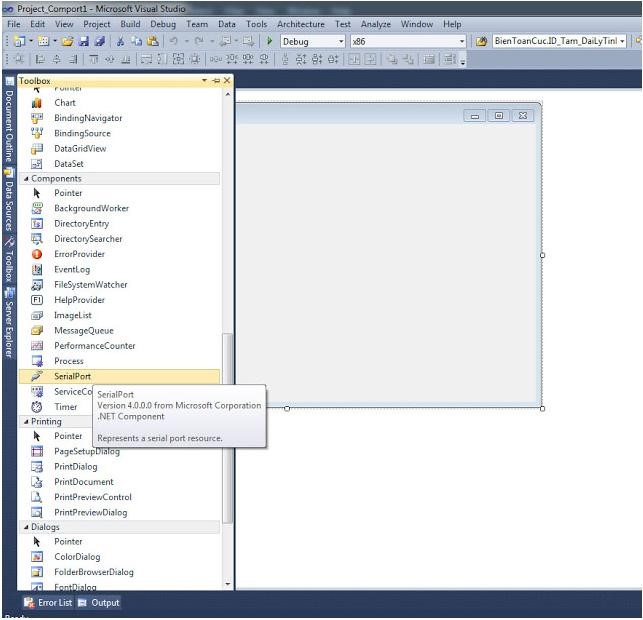
Đầu tiên ta tạo mới một project: **New -> Project**



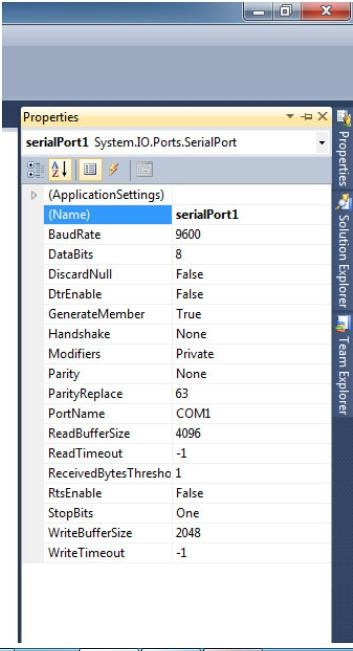
Chọn Windows Forms Application và đặt tên cho chúng. Ở ví dụ dưới đặt là **Project\_comport1**



Kéo lớp SerialPort ở cửa sổ toolbox vào form mới vừa được tạo (Nếu ta không thấy cửa sổ toolbox ta vào menu **View -> Toolbox**).



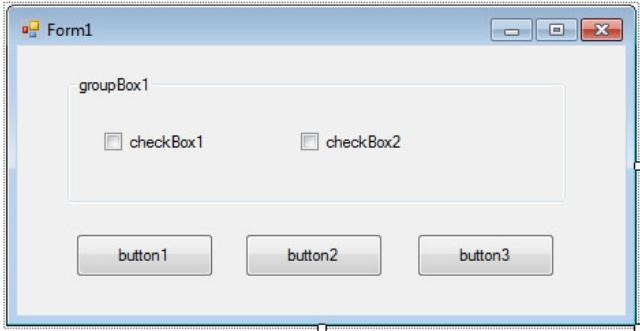
Nhấp chuộc phải lên control vừa mới kéo thả , chọn **Properties** như hình dưới:



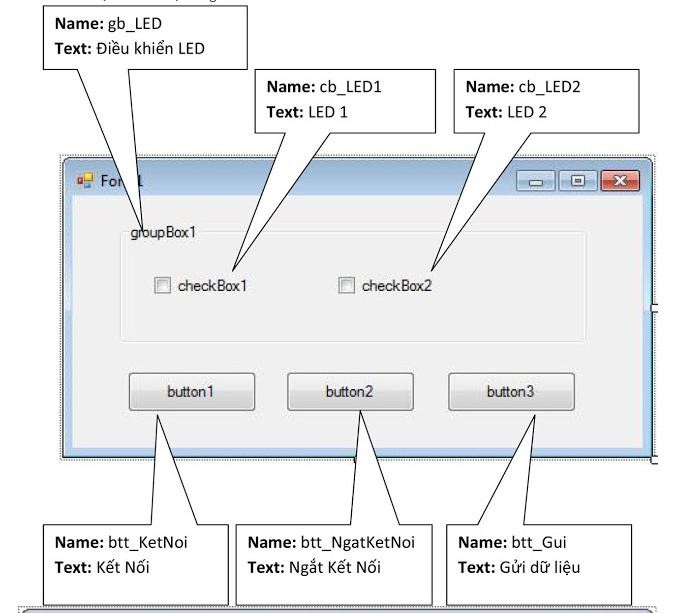
Ta thiết lập lại các thông số của properties serialPort1 như sau:

#### BaudRate : 9600 DataBits : 8 Parity : None StopBits : 1

Thêm vào các control trong cửa sổ toolbox để được như hình bên dưới:

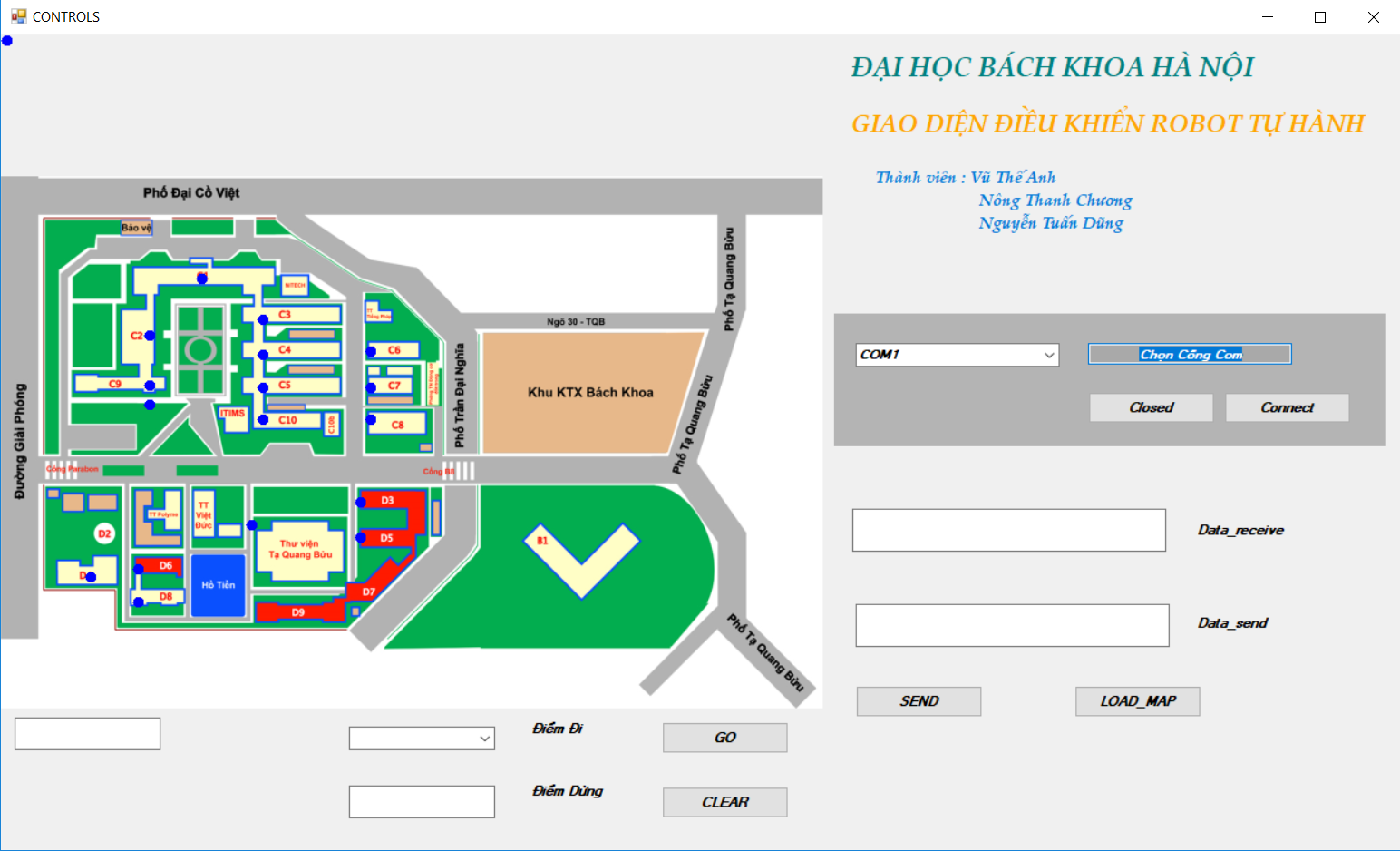


Ta tiến hành đặt tên lại tên (Name) và nhãn hiển thị (Text) cho mỗi control bằng cách nhấn chuộc phải lên từng control (Ví dụ nút nhấn) và chọn Properties



Từ các bước khởi đầu ta có thể lần lượt tạo giao diện mong muốn

Giao diện cuối cùng sau khi thiết kế xong



**Giải pháp**

Thuật toán Dijkstra :

**Giải thuật Dijkstra**, mang tên của 1 nhà khoa học máy tính người Hà Lan **Edsger W. Dijkstra**, là một thuật toán giải quyết bài toán đường đi ngắn nhất trong một đồ thị có hướng **không có cạnh trọng số âm**. Ứng dụng lớn nhất của thuật toán này là trong công nghệ Hệ thống định vị toàn cầu (GPS).

[](http://trachanhso.net/wp-content/uploads/2015/11/edsger_w_dijkstra.jpg)Edsger W. Dijkstra – nhà khoa học máy tính người Hà Lan

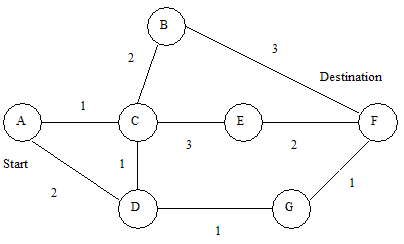
Cho 1 đồ thị có hướng **G = (V, E)** với các cạnh có trọng số không âm, có dữ liệu nhập vào là ma trận trọng số **L** và 2 đỉnh **x, y** cho trước. Việc ta cần làm là tìm đường đi ngắn nhất từ **x** đến **y** trong đồ thị **G**.

Việc chúng ta cần làm là chỉ ra đỉnh **v** bất kì sao cho **x -> v** là đường đi ngắn nhất. Ta gọi **length[v]** là giá trị đường đi ngắn nhất từ **x -> v**, có thể hiểu **length[v]** là giá trị đường đi ngắn nhất trong các đường đi từ đỉnh **x** qua các đỉnh trong tập hợp S (nếu có) rồi đến **v**.

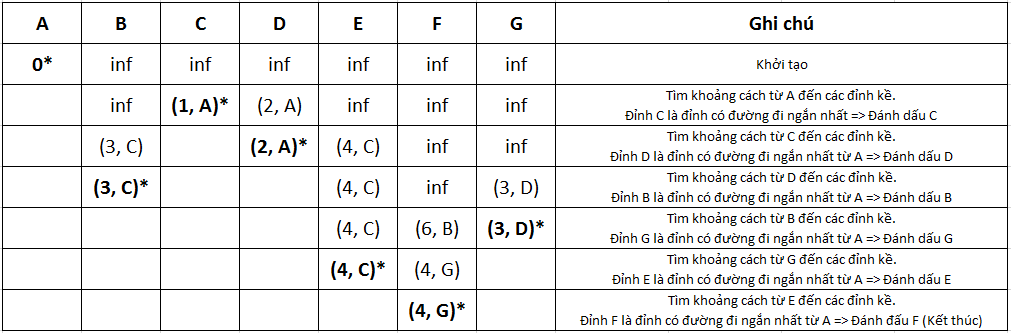
Thuật toán:

1. Khởi tạo các mảng **n**phần tử: **label, length, prev**. Gán **label[k] = 1, length[k] = -1 (inf), prev[k] = -1** với **k** chạy từ **0 -> n – 1**. Gán **length[first] = 0**
2. Chọn đỉnh **v** trong mảng sao cho **length[k]** là nhỏ nhất. Sau đó gán **label[k] = 0**(Đã đánh dấu)
3. Tạo vòng lặp với biến chạy **k**, xét nếu **label[k] = 1**(Chưa đánh dấu) và có đường đi từ **v -> k**: Nếu **length[k] > length[v] +**trọng số từ **v -> k** hoặc **length[k] = inf**, có nghĩa là nếu ta tìm được 1 đường từ **v -> k** là nhỏ nhất, hoặc là chưa tìm được đường nào ngắn nhất (inf) => Gán **length[k] = length[v] +**trọng số **v -> k**, **prev[k] = v**(Tạo vết chân đỉnh trước đó).
4. Nếu **label[last] = 0** (Đã đánh dấu đỉnh đến), kết thúc vòng lặp. Nếu không thì quay lại bước 2.

VD: Ta có 1 đồ thị như sau

[](http://trachanhso.net/wp-content/uploads/2015/11/dijkstras-graph.gif)Đồ thị đã cho.

Ta cần chỉ ra đường đi ngắn nhất từ đỉnh **A** tới **F**. Vậy các bước sẽ như thế nào?

[](http://trachanhso.net/wp-content/uploads/2015/11/record-dijkstra.png)Bảng thống kê các bước thực hiện.

Về nguyên lý, thuật toán Dijkstra giống như việc chạy thi đua vậy. Từ điểm đến ban đầu, ta sẽ mỗi người chạy đến điểm kết thúc theo các đường đi khác nhau. Nếu người nào chạy tới đích trước (tìm min và đánh dấu điểm kết thúc sớm nhất) thì ta xuất ra các đỉnh mà người đó đã đi qua.

**Truyền Thông**

**I.Giao Thức tryền thông(wifi)**

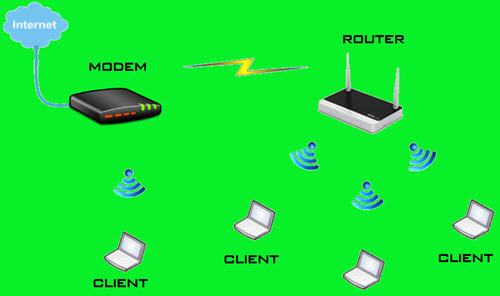
1. WIFI LÀ GÌ.

**WiFi**là viết tắt của **Wireless Fidelity**là phương thức **kết nối không dây sử dụng sóng vô tuyến** .

1. NGUYÊN LÝ HOẠT ĐỘNG CỦA WIFI

Mạng Internet sẽ được các ISP (nhà cung cấp dịch vụ Internet - Internet Services Provider) truyền đến bộ giải mã tín hiệu số (Modem), thông qua bộ định tuyến (Router) hay chúng ta hay gọi là “bộ phát wifi” chuyển tín hiệu hữu tuyến thành kết nối vô tuyến và đưa đến các thiết bị di động không dây thông qua chuẩn kết nối WiFi.

Các thiết bị không dây tiếp nhận sóng WiFi thông qua một thiết bị chuyển đổi tín hiệu gọi là Adapter (card Wifi) được cài đặt trực tiếp trên các thiết bị. Tín hiệu vô tuyến sẽ được giải mã ngay trên thiết bị, từ đây người dùng có thể trực tiếp truy cập Internet như bình thường.



1. MỘT SỐ CHUẨN KẾT NỐI WIFI PHỔ BIẾN.

**Chuẩn 802.11a** : Tần số 5 GHz, tốc độ xử lý 54 Mbps

**Chuẩn 802.11b** : Tần số 2.4 GHz, tốc độ xử lý 11 Mbps, đây là chuẩn kết nối yếu nhất

**Chuẩn 802.11g** : Tần số 2.4 GHz, tốc độ xử lý 54 Mbps

**Chuẩn 802.11n** : Tần số 2.4 GHz, tốc độ xử lý 300 Mbps

1. Các khái niệm cơ bản về WiFi
   1. Station vả Access Point

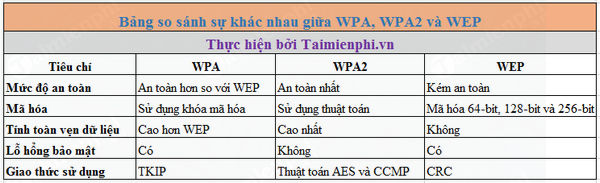
Thiết bị kết nối vào mạng WIFI được gọi là station (trạm). Việc kết nối vào mạng Wifi được hỗ trợ bởi một access point (AP), một AP có chức năng như một hub nhưng dùng cho nhiều station. Một access point thông thường được kết nối vào một mạng dây để phát WIFI (tức là chuyển từ mạng dây sang WIFI). Do đó access point luôn được tích hợp vào router. Mỗi access point được nhận biết bằng một SSID (Service Set IDentifier), SSID cũng là tên của mạng hiển thị khi ta kết nối vào WIFI..

* 1. **Hotspot.**

Một hotspot là một nơi mà các thiết bị có thể kết nối Internet, và thường là bằng WiFi, thông qua mạng WLAN (wireless local area network: mạng nội bộ không dây) nối với router.

* 1. **Các chuẩn bảo mật WiFi**

**WEP (Wired Equivalent Privacy)  
 WPA hay còn gọi Wifi Protected Access  
 WPA2**



**II.Module tryền thông(ESP8266)**

1. Giới thiệu các phiên bản ESP8266(ESP-12).



ESP8266 là một  dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được

1. Tính năng.

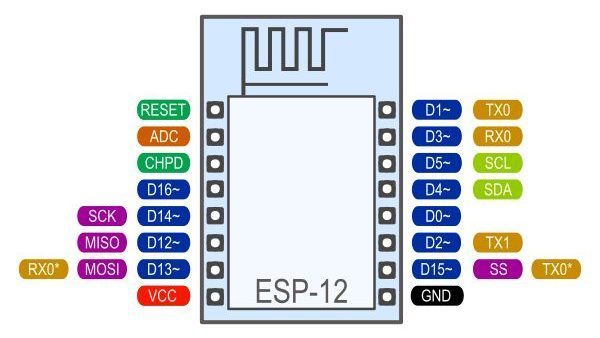
Sử dụng nguồn 3.3v

Tích hợp anten PCB trace trên module

Tiêu chuẩn wifi : 802.11b/g/n, với tần số 2.4GHz và hổ trợ bảo mật WPA/WPA2

Khoảng cách giữa các chân 2mm

1. Sơ đồ chân.



1. Kết nối vào mạng wifi.

| **Lệnh AT** | **Mô tả** | **Thông số** | **Thao tác thực hiện** | **Kết quả trả về** |
| --- | --- | --- | --- | --- |
| AT+CWMODE ="mode" | Cài đặt chế độ | 1 = Station 2 = Access Point  3 = Both | Gõ AT+CWMODE=3 | OK |
| AT+CIPMUX ="mode" | Cài đặt số lượng các kênh kết nối | 0 = 1 kênh kết nối 1 = Nhiều kênh kết nối | Gõ AT+CIPMUX=1 | OK |
| AT+CWLAP | Tìm Wifi |  | Gõ AT+CWLAP | Danh sách các mạng wifi sẵn  có. |
| AT+CWJAP = "ssid", "password" | Kết nối với mạng wifi nhà mình |  | Gõ AT+CWJAP="tên mạng nhà bạn","mật khẩu" | WIFI CONNECTED WIFI GOT IP OK |
| AT+CIFSR | Xem địa chỉ IP của module |  | Gõ AT+CIFSR | +CIFSR:APIP,"192.168.4.1" +CIP:APMAC,"xx:xx:xx:xx:xx:xx" +CIFSR:STAIP,"192.168.1.100" +CIFSR:STAMAC,"xx:xx:xx:xx:xx:xx"  OK |