ĐẠI HỌC QUỐC GIA HÀ NỘI

---

ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO LẬP TRÌNH GHÉP NỐI MÁY TÍNH

Nguyễn Anh Tuấn - 20021594

Đỗ Văn Chí - 18020223

LẬP TRÌNH BỘ ĐẾM SẢN PHẨM TRONG LABVIEW SỬ DỤNG CẢM BIẾN QUANG

HA NOI - 2022

1. **Giới thiệu bài toán**

- Mô tả ngắn gọn về bài toán giải quyết trong đề tài

Tình trạng thiếu lao động đang ngày càng trầm trọng hơn do đại dịch toàn cầu kéo dài, thách thức này tác động to lớn đến hoạt động sản xuất kinh doanh của doanh nghiệp, đặc biệt trong lĩnh vực kho bãi, khi thương mại điện tử ngày càng tăng cao, sự gián đoạn chuỗi cung ứng ảnh hưởng đến trải nghiệm khách hàng.

Đứng trước khó khăn đó, các công ty đã chuyển sang tự động hóa để cân bằng hoạt động hàng ngày, tiết kiệm chi thời gian, chi phí và tạo ra những lợi ích đáng kể. Nhờ áp dụng hệ thống nhà kho thông minh giúp bù đắp các thách thức về lao động hiện tại, giảm thời gian chọn đơn hàng từ 30 phút xuống còn 5 phút và duy trì 99,99% thời gian hoạt động của thiết bị.

- Các vấn đề cần giải quyết liên quan đến lý thuyết đã học

Cảm Biến Hồng ngoại (Chương 2. cảm biến )

Các chế độ đếm (Chương 5. Ghép nối hệ thống đo lường, điều khiển số với máy tính)

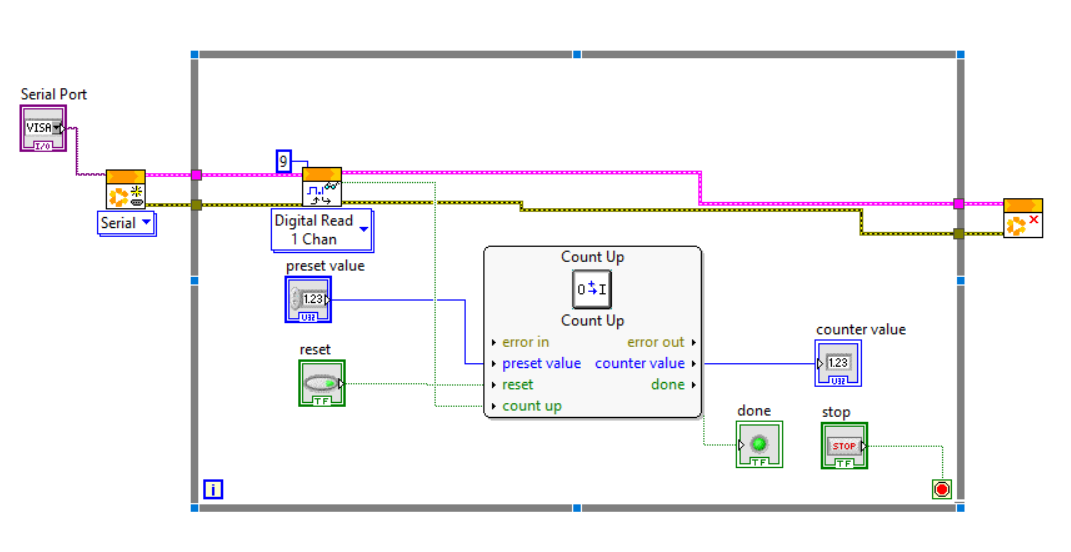
- Các đặc tính cơ bản của sản phẩm

Đếm số lượng sản phẩm

Khi đặt số lượng sản phẩm đặt ra thì đèn sẽ sáng

Khi bấm nút reset thì bộ đếm sẽ bắt đầu lại từ 0

1. **Chức năng khối sản phẩm**

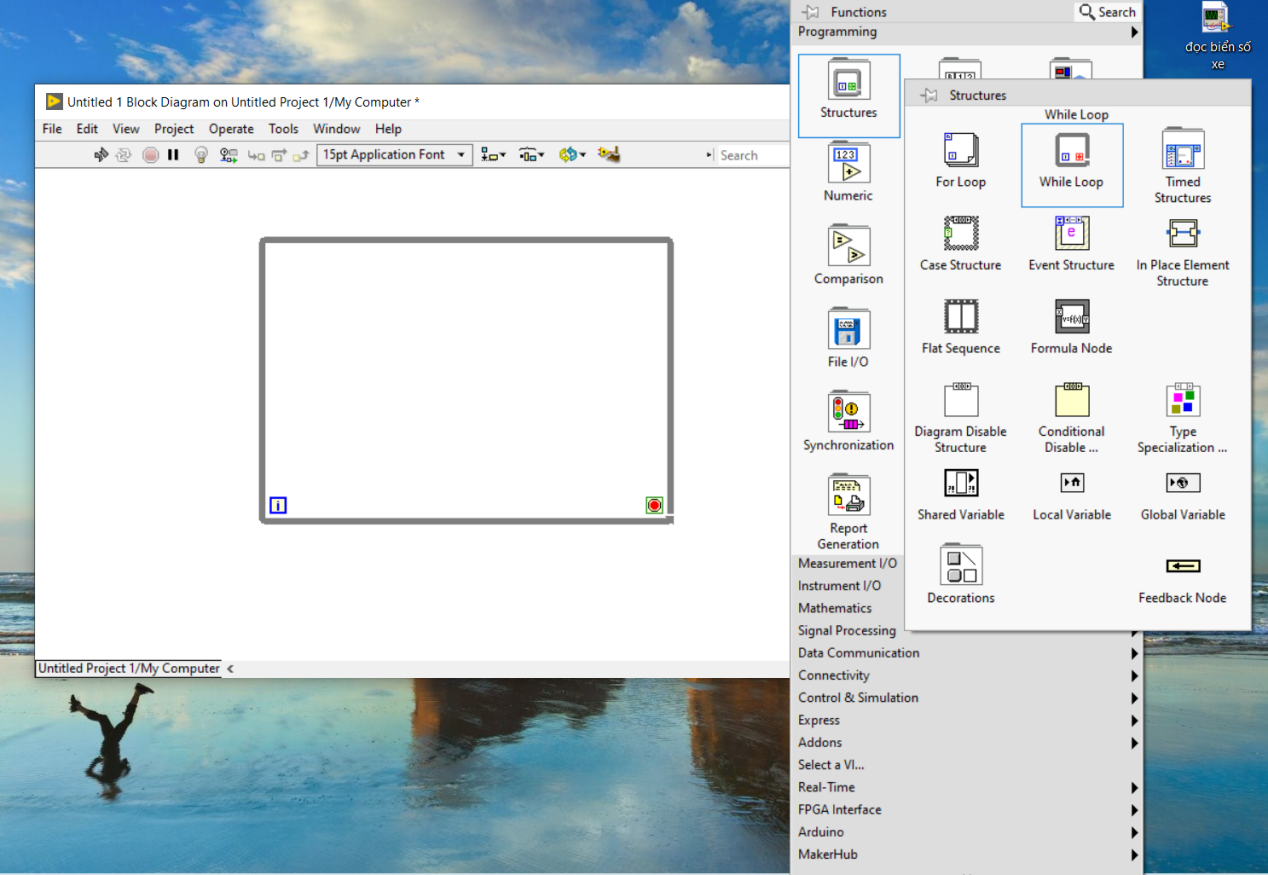
****

**2.1 Vòng lặp While (While Loop)**

Vòng lặp while là vòng lặp có điều kiện . Ý nghĩa của

vòng lặp While là cho phép chạy chương trình mãi tới khi nào nút Stop được nhấn thì mới dừng lại.

Express>> structures >> while loop



-----------------------------------------------------------------------------------------

# 2.2 Open Serial

Open a serial connection to a remote LINX device.

IMG_256

 Makehub >> Linx >>open

#### Inputs

**Serial Port**  
Specifies the COM port of the remote LINX device.

**Baud Rate Override**(Tùy chọn) Khi được chỉ định, cho phép người dùng ghi **đè tốc độ truyền** dfault được sử dụng cho giao tiếp nối tiếp. Theo mặc định, LINX thiết lập giao tiếp bằng cách sử dụng tốc độ truyền 9600 và sau đó thương lượng cho tốc độ truyền được hỗ trợ tối đa của thiết bị LINX từ xa và thiết lập lại giao tiếp ở tốc độ tối đa đó. Nếu đầu vào Ghi đè tốc độ truyền được cung cấp, nó sẽ là tốc độ truyền tối đa được sử dụng. Điều này rất hữu ích khi sử dụng các trừu tượng UART không thể thay đổi tốc độ truyền khi đang di chuyển (ví dụ: Xbee).

**Error In**  
Mô tả các điều kiện lỗi xảy ra trước khi nút này chạy. Đầu vào này cung cấp lỗi tiêu chuẩn trong chức năng.

#### Outputs

**LINX Resource**  
Chứa tài nguyên kết nối LINX.

**Device Name**  
The name of the remote LINX device.

**Error Out**  
Thông tin lỗi. Đầu ra này cung cấp chức năng lỗi tiêu chuẩn.

-------------------------------------------------------------------------------------------------------

# 2.3 Close

Đóng kết nối với thiết bị LINX từ xa và giải phóng mọi tài nguyên I/O cục bộ.

IMG_256

Makehub>> Linx>> close

#### Inputs

**LINX Resource**  
Chứa tài nguyên kết nối LINX.

**Error In**  
Mô tả các điều kiện lỗi xảy ra trước khi nút này chạy. Đầu vào này cung cấp lỗi tiêu chuẩn trong chức năng.

#### Outputs

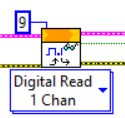
**Error Out**  
Chứa thông tin lỗi. Đầu ra này cung cấp chức năng loại bỏ lỗi tiêu chuẩn.

-----------------------------------------------------------------------------------------

# 2.4 Digital Read 1 Chan

Read the value of the specified digital input channel.

IMG_256

makehub>>linx>>peripheras>>digital>>read

#### Inputs

**LINX Resource**  
Chứa tài nguyên kết nối LINX.

**DI Channel**  
Chỉ định kênh đầu vào kỹ thuật số để đọc từ đó.

**Error In**  
Mô tả các điều kiện lỗi xảy ra trước khi nút này chạy. Đầu vào này cung cấp lỗi tiêu chuẩn trong chức năng.

#### Outputs

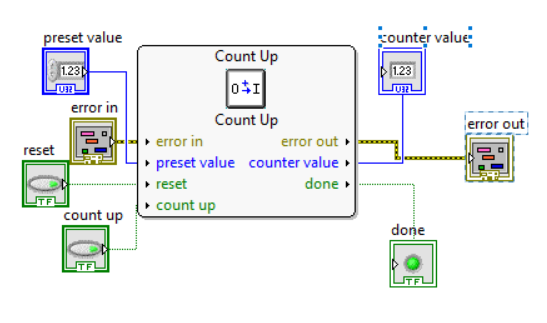
**LINX Resource**  
Chứa tài nguyên kết nối LINX.

**DI Value**  
Returns the digital input value.

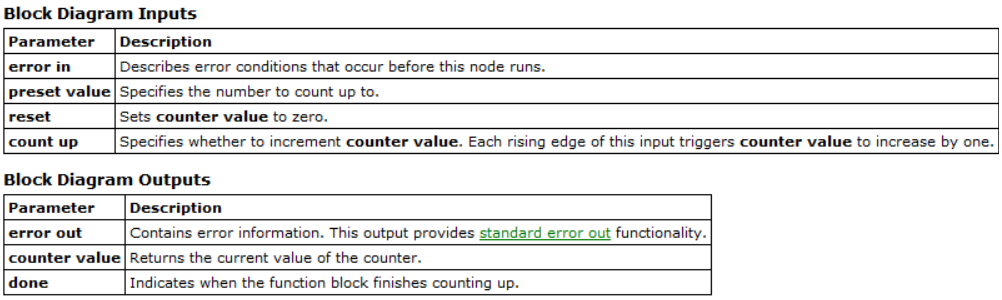
**Error Out**  
Chứa thông tin lỗi. Đầu ra này cung cấp chức năng loại bỏ lỗi tiêu chuẩn.

-----------------------------------------------------------------------------------------

# 2.5 Count Up Function Block



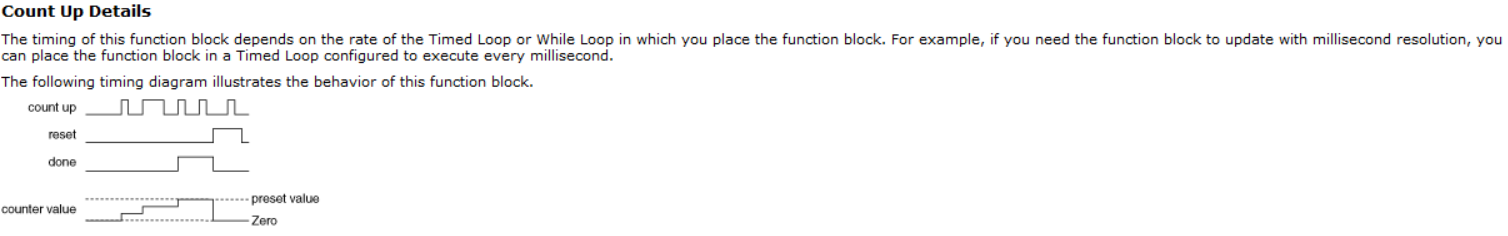
Real time>> Function Block >> Timer>>cout up



## Count Up Details

The timing of this function block depends on the rate of the Timed Loop or While Loop in which you place the function block. For example, if you need the function block to update with millisecond resolution, you can place the function block in a Timed Loop configured to execute every millisecond.

The following timing diagram illustrates the behavior of this function block.



-----------------------------------------------------------------------------------------

2.6. Phần cứng

A. Arduino Uno



## ****Các chân năng lượng****

* **GND (Ground)**: cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
* **5V**: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
* **3.3V**: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
* **Vin (Voltage Input)**: để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
* **IOREF**: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
* **RESET**: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

## ****Bộ nhớ****

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

* **32KB bộ nhớ Flash**: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
* **2KB cho SRAM** (**S**tatic **R**andom **A**ccess **M**emory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
* **1KB cho EEPROM**(**E**lectrically **E**raseble **P**rogrammable **R**ead **O**nly **M**emory): đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

## ****Các cổng vào/ra****

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.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

B. Cảm biến hồng ngoại

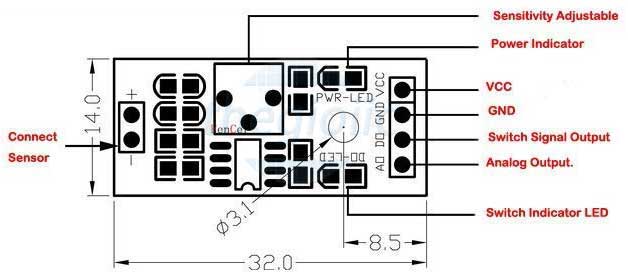
## Tính năng

[LM393](https://www.thegioiic.com/products/lm393-mach-cam-bien-hong-ngoai-phat-hien-lua) là một mạch cảm biến hồng ngoại có thể phát hiện ngọn lửa hoặc bước sóng trong phạm vi 760nm đến 1100nm của nguồn sáng, góc quét 60 độ, khoảng cách phát hiện ngọn lửa khoảng 80cm, ngọn lửa càng lớn, khoảng cách phát hiện càng lớn.

[Cảm biến](https://www.thegioiic.com/product/mach-cam-bien) ngọn lửa nhạy cảm nhất với ánh sáng thông thường đó cũng là một phản ứng, thường được sử dụng làm báo động cháy và các mục đích khác.

**Thông số kỹ thuật:**

* Nguồn: 3.3-5VDC
* Ngõ ra: Analog hoặc Digital
* Kích thước: 32x14mm



1. **Kết quả thực nghiệm, chạy thử, đánh giá.**

- Kết quả mô hình hoạt động tốt trong những tính năng, dự định được đề ra.

- Tiến hành xây dựng mô hình sản phẩm cứng thành công, điều khiển lập trình labview kết nối tốt.

- Dựa trên kết quả trên thực tế đánh giá 9/10 do còn chưa khai thác được nhiều tính năng khác như nhận diện hình ảnh có thể cải tiến và áp dụng để dung trong đề tài lớn hơn.