

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA ĐIỆN- ĐIỆN TỬ



HCMUTE

ĐỒ ÁN MÔN HỌC 2

**THIẾT KẾ HỆ THỐNG ĐO KÍCH
THƯỚC VẬT THỂ HÌNH HỘP VÀ
ĐẾM SẢN PHẨM**

SVTH:

NGUYỄN VĂN LÂN MSSV: 19161254

VÕ HOÀNG QUỐC MSSV: 19161278

Khóa: 19

Ngành: ĐIỆN TỬ- VIỄN THÔNG

GVHD: CÔ PHẠM THỊ THU HIỀN

TP. HỒ CHÍ MINH- 12/2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN-ĐIỆN TỬ
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ VIỄN THÔNG

ĐỒ ÁN MÔN HỌC 2

ĐỀ TÀI
THIẾT KẾ HỆ THỐNG ĐO KÍCH THƯỚC VẬT THỂ
HÌNH HỘP VÀ ĐẾM SẢN PHẨM

Giảng viên hướng dẫn: Cô PHẠM THỊ THU HIỀN

Sinh viên thực hiện: NGUYỄN VĂN LÂN 19161254

VÕ HOÀNG QUỐC 19161278

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến cô ThS. Phạm Thị Thu Hiền đã trực tiếp hướng dẫn, tận tình giúp đỡ, chỉ dạy và tạo điều kiện cho chúng em có thể hoàn thành đồ án môn học 2. Bên cạnh đó, chúng em cũng xin cảm ơn quý thầy cô Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã truyền đạt cho chúng em những kiến thức cần thiết trong suốt thời gian học tập tại trường, nhờ đó chúng em có thể hoàn thành tốt đề tài này. Ngoài ra, em xin cảm ơn đến các bạn sinh viên lớp 19161KTVMA đã hỗ trợ, chia sẻ, trao đổi về mặt kiến thức, xin cảm ơn gia đình đã luôn bên cạnh động viên trong suốt thời gian em thực hiện đề tài. Một lần nữa, chúng em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	iv
DANH MỤC BẢNG	v
DANH MỤC TỪ VIẾT TẮT	vi
Chương 1 TỔNG QUAN.....	1
1.1. Đặt vấn đề.....	1
1.2. Mục tiêu nghiên cứu của đề tài.....	1
1.3. Đối tượng và phạm vi nghiên cứu	1
1.4. Bố cục đề tài	1
Chương 2 CƠ SỞ LÝ THUYẾT	3
2.1. Giới thiệu Arduino UnoR3	3
2.1.1. Sơ lược về Arduino Uno R3	3
2.1.2. Vi điều khiển ATmega328P	5
2.2. Giới thiệu cảm biến khoảng cách hồng ngoại SHARP GP2Y0A21YK0F	6
2.2.1. Giới thiệu chung	6
2.2.2. Thông số kỹ thuật	6
2.2.3 Nguyên lý hoạt động	7
2.3. Giới thiệu module I2C cho LCD	8
2.3.1. Giới thiệu về chuẩn truyền thông I2C	8
2.3.2. Module I2C cho LCD	9
2.5. Giới thiệu về động cơ băng chuyền mini và nguồn tổ ong 24V-5A	11
Chương 3 THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....	13
3.1. Yêu cầu của hệ thống	13
3.2. Phương án thiết kế.....	13
3.3. Sơ đồ khối của hệ thống	14
3.4. Thiết kế phần cứng	16
3.4.1. Khối cảm biến.....	16
3.4.2. Khối hiển thị	16
3.4.3. Khối xử lý trung tâm	18
3.4.4. Khối băng chuyền.....	19

3.4.5. Khôi nguồn	19
3.4.5. Sơ đồ nguyên lý toàn hệ thống	20
3.4.6. Lưu đồ giải thuật	21
Chương 4 KẾT QUẢ THỰC HIỆN	25
4.1 Phân cứng của hệ thống sau khi thiết kế	25
4.2 Vật hộp mẫu để đánh giá hoạt động hệ thống	27
4.3 Khi hộp mẫu để thẳng hàng với băng tải.....	27
4.4 Khi hộp mẫu để nghiêng so với băng tải	30
Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	32
5.1. Kết luận.....	32
5.2. Hướng phát triển.....	32
TÀI LIỆU THAM KHẢO	33
PHỤ LỤC	34

DANH MỤC HÌNH ẢNH

Hình 2.1 Arduino Uno R3	3
Hình 2.2 Sơ đồ chân của Arduino Uno R3.....	4
Hình 2.3 Chip Atemega328	5
Hình 2.4 Cảm biến GP2Y0A21YK0F	6
Hình 2.5 Sơ đồ nguyên lý cảm biến vật cản hồng ngoại	7
Hình 2.6 Cách thức giao tiếp trong I2C	8
Hình 2.7 Module I2C cho LCD	9
Hình 2.8 Bảng chuyển mini 50x6 cm	11
Hình 2.9 Nguồn tổ ong 24V-5A	12
Hình 3.1 Sơ đồ khối hệ thống	14
Hình 3.2 Sơ đồ kết nối các chân tín hiệu các cảm biến	16
Hình 3.3 Sơ đồ kết nối LCD với Arduino Uno R3	17
Hình 3.4 Khối xử lý trung tâm	18
Hình 3.5 Khối băng chuyển	19
Hình 3.6 Sơ đồ nguyên lý toàn hệ thống	20
Hình 3.7 Lưu đồ giải thuật phần 1	21
Hình 3.8 Lưu đồ giải thuật phần 2	22
Hình 3.9 Lưu đồ giải thuật phần 3	23
Hình 4.1 Kết nối các linh kiện phần cứng theo sơ đồ nguyên lý của hệ thống	25
Hình 4.2 Kết nối các linh kiện phần cứng theo sơ đồ nguyên lý của hệ thống	25
Hình 4.3 Hệ thống thực tế hoàn chỉnh	26
Hình 4.4 Vật hộp mẫu dùng để khảo sát	27
Hình 4.5 Giá trị kích thước reset khi không có hộp đi qua hệ thống	28
Hình 4.6 Giá trị kích thước reset khi không có hộp đi qua hệ thống	30

DANH MỤC BẢNG

Bảng 2.1 Đặc tính Arduino.....	3
Bảng 2.2 Các chân kết nối của Arduino Uno R3	4
Bảng 2.3 Thông số kỹ thuật của module I2C cho LCD	9
Bảng 2.4 Chức năng các chân LCD1602	10
Bảng 4.1 Kết quả khảo sát hộp 6x6x5	28
Bảng 4.2 Kết quả khảo sát hộp 7x5x4	29
Bảng 4.3 Kết quả khảo sát hộp 6x6x5 đặt nghiêng	30
Bảng 4.4 Kết quả khảo sát hộp 7x5x4 đặt nghiêng	31

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ đầy đủ
1	I2C	Inter-integrated circuit
2	LCD	Liquid Crystal Display
3	ROM	Read-Only Memory
4	RAM	Random Access Memory
5	UART	Universal Asynchronous Receiver Transmitter
6	SPI	Serial Peripheral Interface
7	GPIO	General-purpose input/output
8	USB	Universal Serial Bus
9	PWM	Pulse Width Modulation
10	SDA	Serial Clock Line
11	SCL	Serial Data Line

Chương 1 TỔNG QUAN

1.1. Đặt vấn đề

Xã hội phát triển kéo theo nhu cầu của con người ngày càng lớn đặc biệt là những nhu cầu về giảm bớt tải trọng trong lao động để có một cuộc sống ngày càng tốt đẹp và đầy đủ tiện nghi hơn. Và từ đó những công nghệ hiện đại ra đời dần dần thay thế con người trong nhiều lĩnh vực để tăng năng suất lao động. Những ưu điểm của máy móc tự động trong công nghiệp đó là vận hành một cách chính xác, rất ít khi xảy ra sai sót, chi phí ban đầu lớn nhưng sử dụng lâu dài và đem lại lợi nhuận cao. Trong một doanh nghiệp sản xuất hàng hóa ở khâu đóng gói sản phẩm hầu hết các sản phẩm được đựng gói lại trong thùng chứa hình hộp do đó yêu cầu về đo kích thước của vật hộp đó để phù hợp với kích thước sản phẩm để tối ưu kinh phí cho những thứ dùng để đóng gói sản phẩm. Các doanh nghiệp đó không thể nào thuê nhân công để đo từng cái hộp vì như vậy sẽ rất tốn thời gian và chi phí thuê nhân công. Vì thế yêu cầu đặt ra là có một hệ thống có thể đo được kích thước của những vật hộp đó. Như vậy, để có thể hiểu thêm về hệ thống tự động nói chung, các bước thiết kế cũng như lợi ích của hệ thống đo kích thước vật thể hình hộp nói riêng trong thực tế, người thực hiện chọn đề tài “Thiết kế hệ thống đo kích thước vật thể hình hộp”.

1.2. Mục tiêu nghiên cứu của đề tài

Vận dụng các kiến thức về vi xử lý, ngôn ngữ lập trình C, các kiến thức về cảm biến, băng chuyền mini, LCD, chuẩn giao tiếp I2C để thực hiện nghiên cứu đề tài thiết kế hệ thống đo kích thước vật thể hình hộp về chiều dài, rộng và cao.

1.3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu là vi điều khiển, LCD, module cảm biến khoảng cách hồng ngoại SHARP GP2Y0A21YK0F, băng chuyền mini, các chuẩn truyền thông không dây và có dây.

Phạm vi nghiên cứu là thiết kế hệ thống đo kích thước đo vật thể hình hộp gồm chiều dài, rộng, cao và đếm số lượng sản phẩm sau đó hiển thị lên LCD.

1.4. Bố cục đề tài

Đề tài sẽ được trình bày trong 5 chương:

Chương 1: Tổng quan

Trong chương này trình bày lý do chọn đề tài, các mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài.

Chương 2: Cơ sở lý thuyết

Giới thiệu về board Arduino UNO R3, cảm biến khoảng cách hồng ngoại SHARP

GP2Y0A21YK0F, LCD, bảng chuyển mini, chuẩn truyền I2C và các công nghệ truyền thông không dây phổ biến hiện nay.

Chương 3: Thiết kế và xây dựng hệ thống

Trình bày các yêu cầu đặt ra của hệ thống, xây dựng sơ đồ khối và thiết kế sơ đồ nguyên lý của hệ thống.

Chương 4: Kết quả thực hiện

Trình bày các kết quả và hình ảnh thực tế của hệ thống.

Chương 5: Kết luận và hướng phát triển

Đưa ra kết luận và hướng phát triển của đề tài

Chương 2 CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu Arduino Uno R3

2.1.1. Sơ lược về Arduino Uno R3

Arduino UNO R3 là kit Arduino UNO thế hệ thứ 3, được trang bị cấu hình mạnh cho các loại bộ nhớ ROM, RAM và Flash, các chân GPIO của Arduino Uno R3 có nhiều khả năng khác nhau như: xuất xung PWM, đọc các tín hiệu analog và thực hiện các giao tiếp UART, SPI, TWI (I2C). Vì vậy, Arduino Uno R3 có khả năng lập trình cho những hệ thống phức tạp.



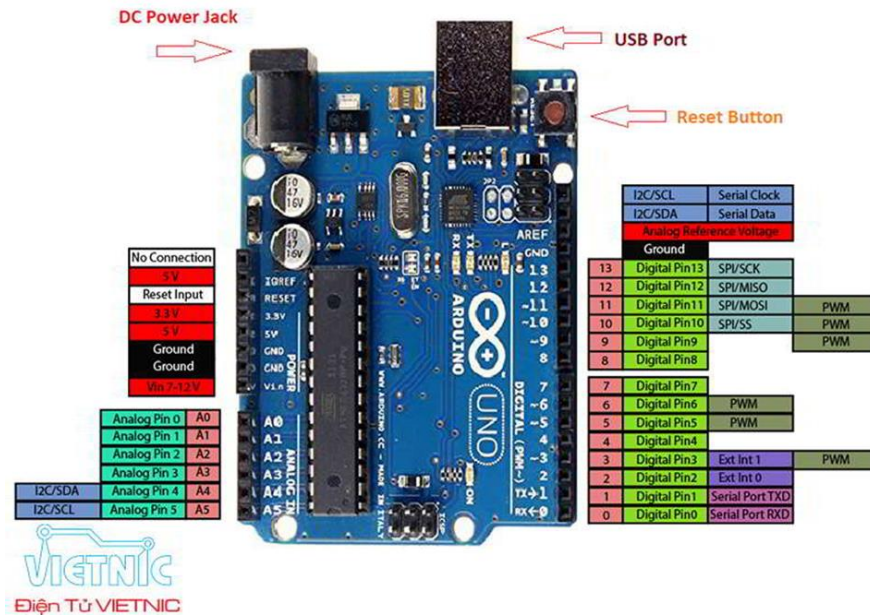
Hình 2.1 Arduino Uno R3

Đặc tính của Arduino Uno R3 được thể hiện ở bảng dưới đây.

Điện áp hoạt động	5V
Tần số hoạt động	16Mhz
Điện áp đầu vào	7-12 VDC
Điện áp giới hạn	6-20VDC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa ở mỗi chân	30mA
Dòng ra tối đa	5V-500mA
Bộ nhớ flash	32KB
SRAM	2KB
EEPROM	1KB
Vi điều khiển	Atmega 328
Tốc độ xung nhịp	16MHz

Bảng 2.1 Đặc tính Arduino

Sơ đồ chân Arduino Uno R3



Hình 2.2 Sơ đồ chân của Arduino Uno R3

Tên chân	Ký hiệu	Chức năng
Chân nguồn	Jack cắm	Cấp nguồn cho Board
Chân USB	Cổng USB	Cổng giao tiếp USB và nạp code
Chân GPIO	Pin0~Pin13	14 chân tín hiệu ngõ vào – ngõ ra dạng số
Chân Analog	A0~A5	Ngõ vào của tín hiệu tương tự
Chân năng lượng	Vin	để cấp nguồn ngoài cho board
	Reset (nút)	Khởi động lại board
	5V	Cấp điện áp đầu ra
	3.3V	
	IOREF	Để đo điện áp hoạt động
Chân Serial	0 (RX) và 1 (TX)	Truyền và nhận dữ liệu
Chân PWM	3, 5, 6, 9, 10, và 11	Xuất ra xung PWM
Chân giao tiếp SPI	10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)	Các chân dùng để giao tiếp SPI

Bảng 2.2 Các chân kết nối của Arduino Uno R3

2.1.2. Vi điều khiển ATmega328P

Atmega328 là một chip vi điều khiển thuộc họ MegaAVR được sản xuất bởi hãng Atmel. Atmega328 hoạt động trong một dải điện áp rộng từ 1.8V – 5.5V, tốc độ thực thi là 1MIPS trên 1MHz.

Thông số chính Atmega328P-PU:

- + Kiến trúc: AVR 8bit
- + Xung nhịp lớn nhất: 20Mhz
- + Bộ nhớ chương trình (FLASH): 32KB
- + Bộ nhớ EEPROM: 1KB
- + Bộ nhớ RAM: 2KB
- + Điện áp hoạt động rộng: 1.8V – 5.5V
- + Số timer: 3 timer gồm 2 timer 8-bit và 1 timer 16-bit
- + Số kênh xung PWM: 6 kênh (1 timer 2 kênh)



Hình 2.3 Chip Atemega328

Ngày nay vi điều khiển Atmega328 thực sự được sử dụng phổ biến từ các dự án nhỏ của sinh viên, học sinh với giá thành rẻ, xử lý mạnh mẽ, tiêu tốn ít năng lượng (chế độ hoạt động: 0.2 mA, chế độ ngủ: 0.1 μ A, chế độ tích kiệm: 0.75 μ A) và sự hỗ trợ nhiệt tình của cộng đồng người dùng AVR. Và không thể không nhắc tới sự thành công của Vi điều khiển Atmega328 trong dự án mã nguồn mở Arduino với các module Aduino Uno (R3), Arduino Nano, Arduino Pro mini những sản phẩm dẫn dắt chúng ta vào thế giới mã nguồn mở để hoàn thành một chương trình một cách nhanh chóng.

2.2. Giới thiệu cảm biến khoảng cách hồng ngoại SHARP GP2Y0A21YK0F

2.2.1. Giới thiệu chung

GP2Y0A21YK0F là bộ cảm biến đo khoảng cách, bao gồm sự kết hợp tích hợp của PSD (máy dò nhảy cảm vị trí), IRED (phát hồng ngoại diode) và mạch xử lý tín hiệu. Sự đa dạng của hệ số phản xạ của vật thể, nhiệt độ môi trường và thời gian hoạt động không bị ảnh hưởng dễ dàng để phát hiện khoảng cách nhờ áp dụng phương pháp tam giác hóa. Thiết bị này xuất ra điện áp tương ứng với khoảng cách phát hiện. Vì vậy, cảm biến này cũng có thể được sử dụng như một cảm biến tiệm cận.

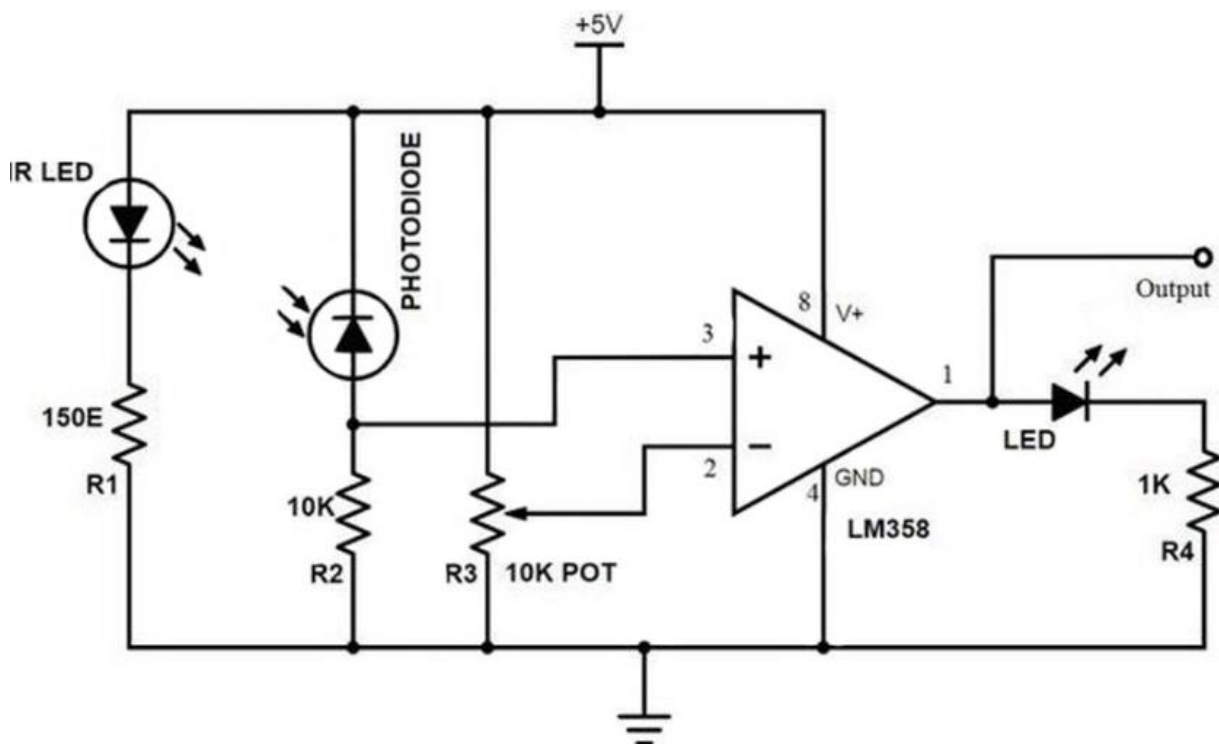


Hình 2.4 Cảm biến GP2Y0A21YK0F

2.2.2. Thông số kỹ thuật

- Model: GP2Y0A21YK0F
- Điện áp sử dụng: 4.5 ~ 5.5VDC.
- Dòng sử dụng trung bình: 30 mA (note: this sensor draws current in large, short bursts, and the manufacturer recommends putting a 10 μ F capacitor or larger across power and ground close to the sensor to stabilize the power supply line)
- Khoảng cách đo: 10 ~ 80cm.
- Dạng tín hiệu trả về: analog voltage
- Kích thước: 29.5×13×13.5mm

2.2.3 Nguyên lý hoạt động



Hình 2.5 Sơ đồ nguyên lý cảm biến vật cản hồng ngoại

Led phát hồng ngoại (IR LED) luôn luôn phát ra sóng ánh sáng có bước sóng hồng ngoại mắt người không thể nhìn thấy ánh sáng này, vì vậy người ta sử dụng led thu hồng ngoại, led thu hồng ngoại bình thường nó có nội trở rất lớn (vài trăm $K\Omega$), khi led thu được tia hồng ngoại chiếu vào đủ lớn thì nội trở của nó giảm xuống (cỡ vài chục Ω).

Khi gặp vật cản, những chùm tia hồng ngoại gặp vật cản và phản xạ lại led thu làm led thu thay đổi giá trị điện trở. Ở đây chúng ta thấy cầu chia áp ở điện trở R2 và mắt thu hồng ngoại, sự thay đổi điện trở của mắt thu hồng ngoại dẫn đến điện áp đầu vào chân 3 Op-Amp cũng thay đổi.

Khi khoảng cách càng gần, sự thay đổi càng lớn.

Khi đó, điện áp đầu vào chân 3 Op-Amp được so sánh với giá trị điện áp không đổi trên biến trở R3, nếu điện áp chân 3 Op-Amp lớn hơn điện áp chân 2 Op-Amp thì Op-Amp xuất mức 1 (bằng VCC) . Ngược lại nếu điện áp chân 3 Op-Amp nhỏ hơn điện áp chân 2 Op-Amp thì Op-Amp xuất mức 0 (bằng GND)

Điện trở như R1 (150 Ω), R2 (10K Ω), R4 (1 K Ω) giúp các led hoạt động mà không bị cháy

Biến trở R3 dùng để chỉnh độ nhạy của biến trở

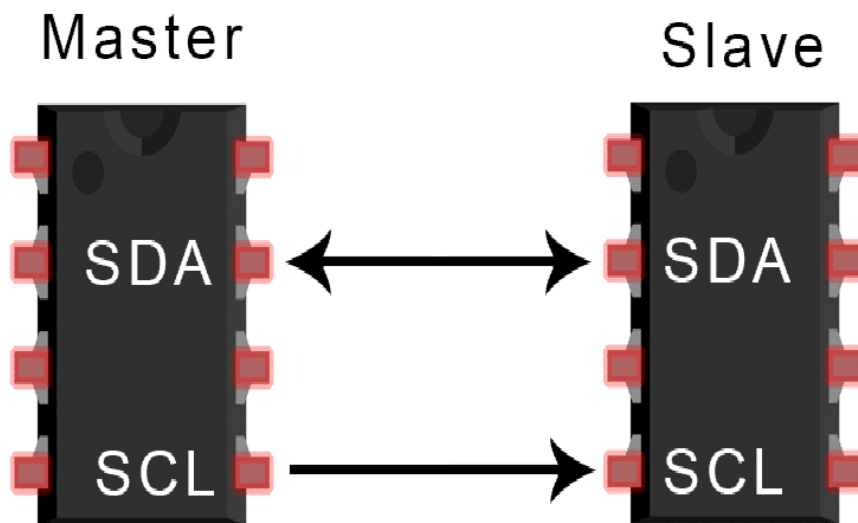
2.3. Giới thiệu module I2C cho LCD

2.3.1. Giới thiệu về chuẩn truyền thông I2C

Chuẩn giao tiếp I2C tên tiếng Anh Inter-Integrated Circuit được phát triển bởi hãng sản xuất linh kiện điện tử Philips dùng cho quá trình giao tiếp giữa các IC với đặc điểm là một loại bus nối tiếp hai chiều với hai dây tín hiệu. Đây là một trong những chuẩn truyền dữ liệu thường gặp nhất hiện nay.

I2C là sự kết hợp các tính năng của SPI và UART. I2C có thể kết nối nhiều dạng khác nhau như: một master với một slave, một master với nhiều slave, nhiều master với một slave và nhiều master với nhiều slave.

Giao tiếp I2C là truyền đồng bộ nối tiếp. I2C có kết nối đơn giản, chi phí thấp, tốc độ truyền nhanh phù hợp cho giao tiếp giữa vi điều khiển và các loại cảm biến, hiển thị, IoT, EEPROMs, v.v ...



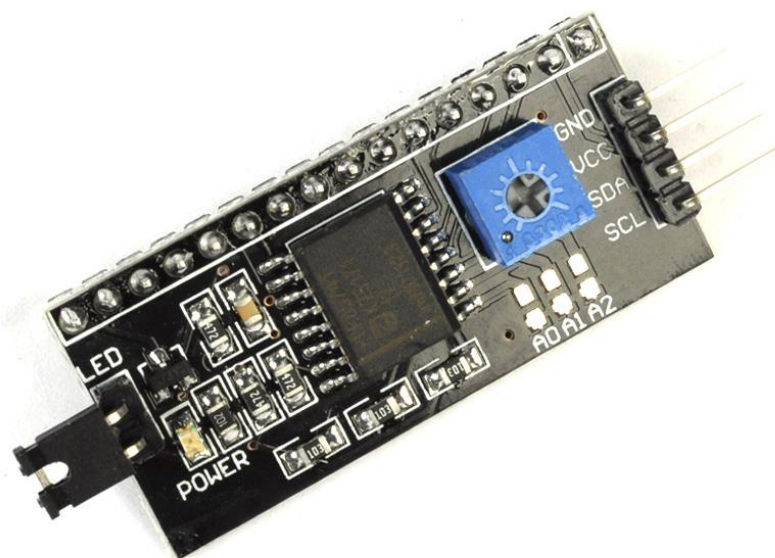
Hình 2.6 Cách thức giao tiếp trong I2C

Tương tự với UART, I2C cũng sử dụng hai dây để truyền dữ liệu giữa các master và slave:

- SDA (Serial Data) – đường dùng để cho master và slave truyền và nhận dữ liệu
- SCL (Serial Clock) - đường mang tín hiệu xung clock.

I2C là một giao thức truyền thông nối tiếp, vì vậy dữ liệu được truyền theo từng bit theo một đường duy nhất là SDA. Tín hiệu xung clock luôn được điều khiển bởi master.

2.3.2. Module I2C cho LCD



Hình 2.7 Module I2C cho LCD

Dưới đây là bảng thông số kỹ thuật của module I2C cho LCD.

Điện áp hoạt động	2.5-6V DC
Hỗ trợ màn hình	LCD1602, 1604, 2004
Giao tiếp	I2C
Địa chỉ mặc định	0X27
Kích thước	41.5mmx19mmx15.3mm
Trọng lượng	5g

Bảng 2.3 Thông số kỹ thuật của module I2C cho LCD

Ngoài ra module còn tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt và tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

2.4. Giới thiệu LCD1602

Ngày nay, LCD1602 (Liquid Crystal Display) được ứng dụng rất nhiều trong các ứng dụng vi điều khiển bởi khả năng hiển thị đa dạng nhiều loại kí: số, chữ, đồ họa,..., dễ dàng lắp ráp và giao tiếp với nhiều hình thức khác nhau, sử dụng ít tài nguyên hệ thống, giá thành rẻ,...

Thông số kỹ thuật của LCD1602:

- Điện áp lớn nhất: 7V
- Điện áp nhỏ nhất: - 0.3V
- Hoạt động ổn định: 2.7-5.5V
- Điện áp ra mức cao: > 2.4
- Điện áp ra mức thấp: <0.4V
- Dòng điện cấp nguồn: 350uA đến 600uA
- Nhiệt độ hoạt động: - 30 đến 75 độ C

Chân	Ký hiệu	Chức năng
1	VSS	Nối đất
2	VDD	Kết nối tới nguồn 3.3VDC
3	VL	Điều chỉnh độ tương phản
4	RS	Chọn thanh ghi
5	R/W	Chân chọn chế độ đọc/ghi
6	E	Chân cho phép LCD hoạt động
7 - 14	D0 – D7	Chân dữ liệu cho LCD
15	BLA	Nguồn của đèn nền LCD – đầu Anode
16	BLK	Nguồn của đèn nền LCD – đầu Kathode

Bảng 2.4 Chức năng các chân LCD1602

2.5. Giới thiệu về động cơ băng chuyền mini và nguồn tổ ong 24V-5A



Liên Hệ: 024.66.88.66.86 - Hotline: 0932.488.998

Hình 2.8 Băng chuyền mini 50x6 cm

Cấu tạo và thông số kỹ thuật:

- Khung được cấu tạo từ nhôm định hình.
- Dây băng màu xanh lá dày 2mm.
- Con lăn đỡ dây băng tải
- Động cơ giảm tốc 24V.
- Bốn chân đỡ làm bằng nhựa.
- Kích thước: 50x6 cm
- Tốc độ băng chuyền: 24V-48RPM

Băng chuyền được điều khiển bởi nguồn đầu vào là nguồn tổ ong 24V-5A



Hình 2.9 Nguồn tổ ong 24V-5A

Nguồn Tổ Ong 24V 5A hay còn gọi là bộ nguồn chuyển đổi điện áp từ 220V AC sang 24V DC để cung cấp điện áp cho hoạt động của các thiết bị. Nguồn tổ ong khá phổ biến trong cuộc sống. Trong công nghiệp nó có thể được dùng để cấp nguồn cho các thiết bị trong tủ điện.

Thông số kỹ thuật:

- Điện áp đầu vào: 110 – 220VAC
- Ngõ ra:
 - + Dòng điện: 5A
 - + Điện áp: 24V
 - + Công suất: 120W
- Kích thước nguồn tổ ong: 400g
- Trọng lượng: 19.8x9.8x4 cm

Chương 3 THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

3.1. Yêu cầu của hệ thống

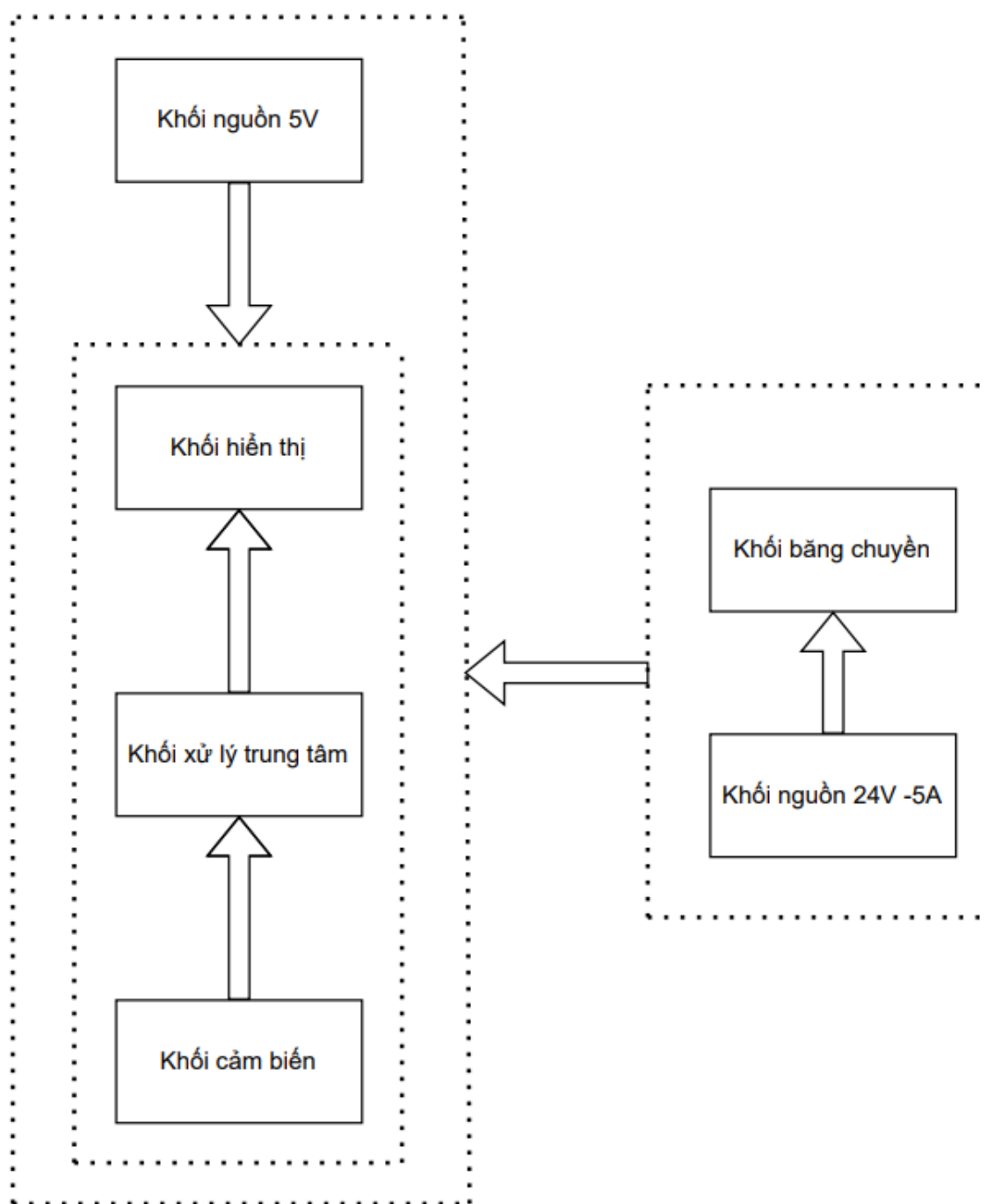
Thiết kế một hệ thống đơn giản có thể đo được kích thước hộp bằng cách sử dụng các cảm biến đo khoảng cách khi vật được đặt lên băng chuyền và đi qua các cảm biến này. Và hệ thống có các chức năng sau:

- Có một băng chuyền để đặt vật lên.
- Có các cảm biến có chức năng đo khoảng cách. Cụ thể là đo các kích thước của hộp là chiều dài, chiều rộng, chiều cao và đo được nhiều dạng hộp với những kích thước khác nhau.
- Có thể đếm được số lượng sản phẩm.
- Có LCD để hiển thị kết quả đo được.

3.2. Phương án thiết kế

- Hệ thống sử dụng vi điều khiển Arduino Uno R3 làm khối xử lý trung tâm để xử lý kết quả đo được từ các cảm biến cũng như hiển thị kết quả lên LCD.
- Sử dụng ba cảm biến đo khoảng cách là SHARP GP2Y0A21YK0F để đo chiều rộng và chiều cao của hộp và ba cảm biến này được thiết kế đặt nằm trong 1 chiếc hộp. Cụ thể là: Hai cảm biến được đặt song song với nhau sao cho bộ phận thu và phát sóng hướng về phía mặt đối diện của hộp dùng để đo chiều rộng và hai cảm biến này được đặt ở 2 mặt đối diện của hộp, một cảm biến được đặt ở mặt phía trên của hộp dùng để đo chiều cao.
- Một mặt của chiếc hộp sẽ gắn LCD1602 lên.
- Sau đó kết nối các cảm biến và LCD1602 với Arduino Uno R3.
- Về phần đo chiều dài thì em sẽ tính toán dựa trên tốc độ của băng chuyền (24V- 48RPM) mà nhà sản xuất băng chuyền đã cung cấp sẵn kết hợp với thời gian thay đổi của một trong hai kích thước còn lại, sau đó sử dụng công thức $s=v.t$ để tính ra được chiều dài.
- Các kích thước lý thuyết của hộp chứa cảm biến và LCD cũng như các mô hình hộp để kiểm chứng sau này sẽ được cho trước sau đó kiểm chứng và nhận xét độ chính xác
- Phần tiếp theo sẽ là vẽ lưu đồ giải thuật cũng như viết chương trình.
- Lắp ráp băng chuyền và hộp chứa cảm biến và LCD với nhau để được hệ thống hoàn chỉnh.

3.3. Sơ đồ khối của hệ thống



Hình 3.1 Sơ đồ khối hệ thống

Chức năng của các khối trên sơ đồ khối hình 3.1 là:

- Khối nguồn 5V: Dùng để cấp nguồn DC 5V cho khối xử lý trung tâm, khối hiển thị và khối cảm biến hoạt động
- Khối hiển thị: Là LCD1602 dùng để hiển thị kết quả.
- Khối cảm biến: Có chức năng đo đạc các khoảng cách sau đó gửi về khối xử lý trung tâm để tính toán và xử lý.
- Khối xử lý trung tâm: Xử lý tất cả các tín hiệu từ cảm biến gửi về và xử lý các tín hiệu đó sau đó gửi tín hiệu cho khối hiển thị để hiển thị kết quả.
- Khối nguồn 24V-5A: Dùng để cấp nguồn cho băng chuyền.
- Khối băng chuyền: Dùng để cho vật hộp chạy qua, bên cạnh đó thì băng chuyền còn có chức năng cho biết vận tốc của nó để từ đó tính toán được chiều dài của hộp.

Với các khối chức năng đã nêu thì hệ thống hoạt động như sau: Khi cấp nguồn DC 5V sẽ khởi động ba khối là: khối cảm biến, khối hiển thị và khối xử lý trung tâm. Khối xử lý trung tâm sẽ chờ tín hiệu được gửi về từ khối cảm biến và khối hiển thị sẽ chờ tín hiệu hiển thị từ khối xử lý trung tâm. Cấp nguồn 24V-5A sẽ khởi động băng chuyền. Khi hệ thống vừa khởi động chưa có vật hộp đặt lên băng chuyền thì khoảng cách đo được ở khối cảm biến là một hằng số cố định và đó chính là các khoảng cách mặc định, khối xử lý trung tâm khi nhận các khoảng cách mặc định này sẽ được nhận biết và không phát tín hiệu hiển thị cho khối hiển thị. Khi đặt vật hộp lên băng chuyền hộp sẽ từ từ duy chuyển ngang qua khối cảm biến cụ thể là duy chuyển qua lần lượt các cảm biến dùng để đo chiều rộng và chiều cao, khi đó các khoảng cách đo được ở cảm biến sẽ thay đổi và khối xử lý trung tâm nhận được các tín hiệu khoảng cách mới này sẽ xử lý và phát tín hiệu cho khối hiển thị sau đó khối hiển thị sẽ hiển thị kết quả. Cụ thể là:

- Với đo chiều rộng: Khi vật đi qua lần lượt hai cảm biến đo khoảng cách thì tín hiệu khoảng cách nhận được ở khối xử lý trung tâm sẽ thay đổi, từ đó chiều rộng được xác định bằng cách lấy tín hiệu khoảng cách mặc định ban đầu trừ đi tổng hai tín hiệu khoảng cách mới vừa thay đổi. Ví dụ: Tín hiệu khoảng cách mặc định đo được là 10, tín hiệu khoảng cách sau khi hộp đi qua ở hai cảm biến lần lượt là 3 và 4 thì chiều rộng = $10 - (3 + 4)$.
- Với đo chiều cao: Tương tự như khi đo chiều rộng thì chiều cao được xác định bằng cách lấy tín hiệu khoảng cách mặc định trừ đi tín hiệu khoảng cách thay đổi khi vật đi qua cảm biến dùng để đo chiều cao.

Đối với khối băng chuyền chúng ta sẽ sử dụng tham số tốc độ của băng chuyền và tính khoảng thời gian thay đổi khoảng cách ở chiều cao từ khoảng cách mặc định sang khoảng cách thay đổi khi có hộp đi qua sau đó sử dụng công thức $s = v.t$ để tính toán được chiều dài của hộp.

- Với đếm số lượng sản phẩm: Khi vật đi đặt lên băng chuyền và di chuyển ngang qua các cảm biến lúc này tín hiệu nhận được từ cảm biến ở bộ xử lý trung tâm sẽ thay đổi từ sự

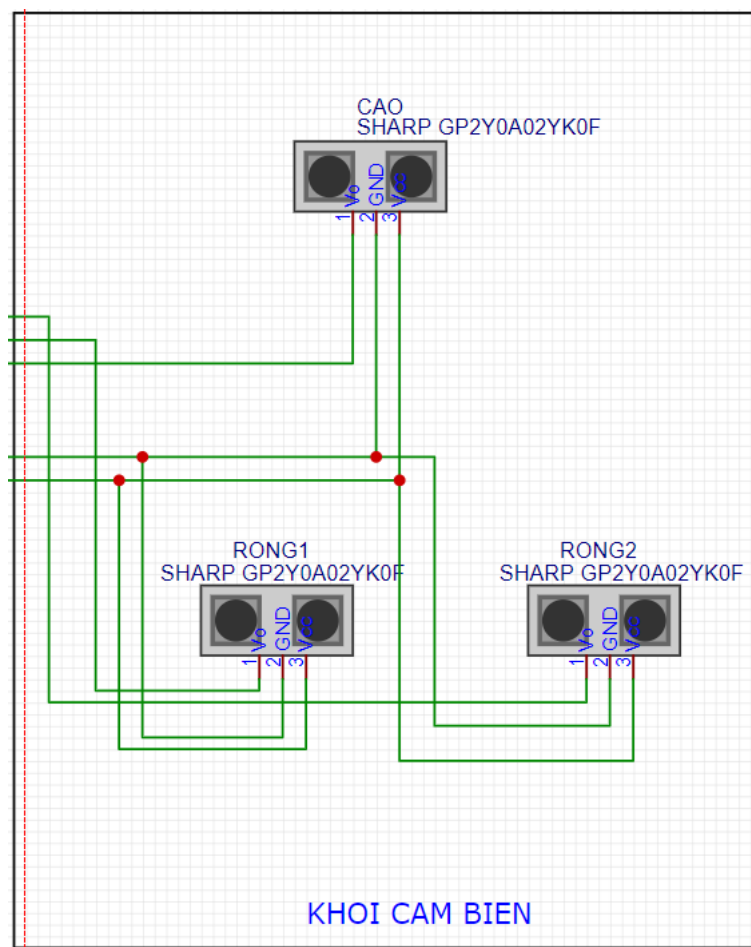
thay đổi đó ta có thể tính được số lượng sản phẩm bằng số lần thay đổi tín hiệu của cảm biến khi gửi về khối xử lý trung tâm.

Các khoảng cách và số lượng sản phẩm sau khi được khối xử lý trung tâm tính toán xử lý xong sẽ phát tín hiệu hiển thị các kích thước lên khối hiển thị.

3.4. Thiết kế phần cứng

3.4.1. Khối cảm biến

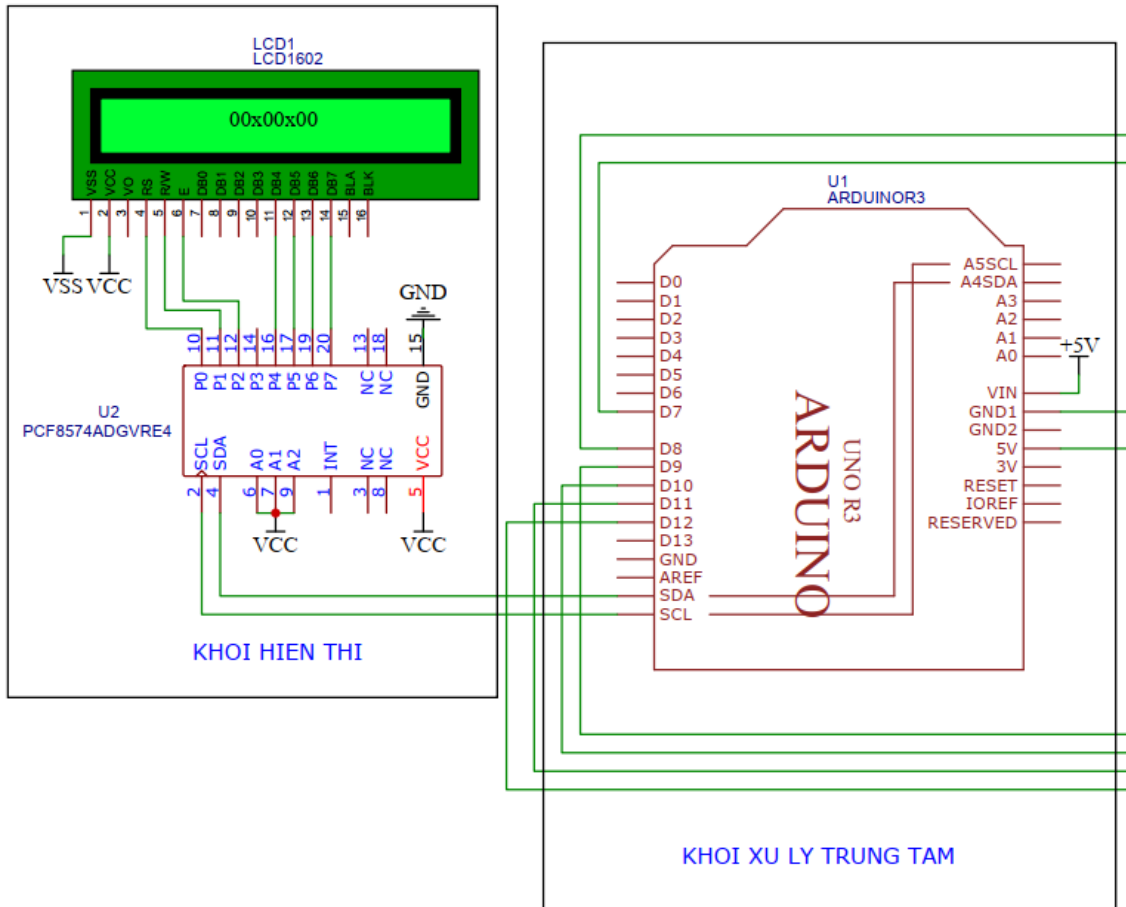
Khối cảm biến sử dụng 3 cảm biến SHARP GP2Y0A02YK0F với chân V_o (chân tín hiệu ra) của các cảm biến CAO, RONG1, RONG2 được kết nối lần lượt với các chân từ A0-A2 của Arduino. Vcc sẽ kết nối với chân 5V và GND kết nối với GND.



Hình 3.2 Sơ đồ kết nối các chân tín hiệu các cảm biến

3.4.2. Khối hiển thị

Khối hiển thị là LCD1602 được kết nối với Arduino Uno R3 thông qua module I2C để hiển thị dễ dàng hơn.



Hình 3.3 Sơ đồ kết nối LCD với Arduino Uno R3

Bên trong module I2C cho LCD là một module tên PCF8574 dùng để mở rộng chân I/O thông qua giao tiếp I2C.

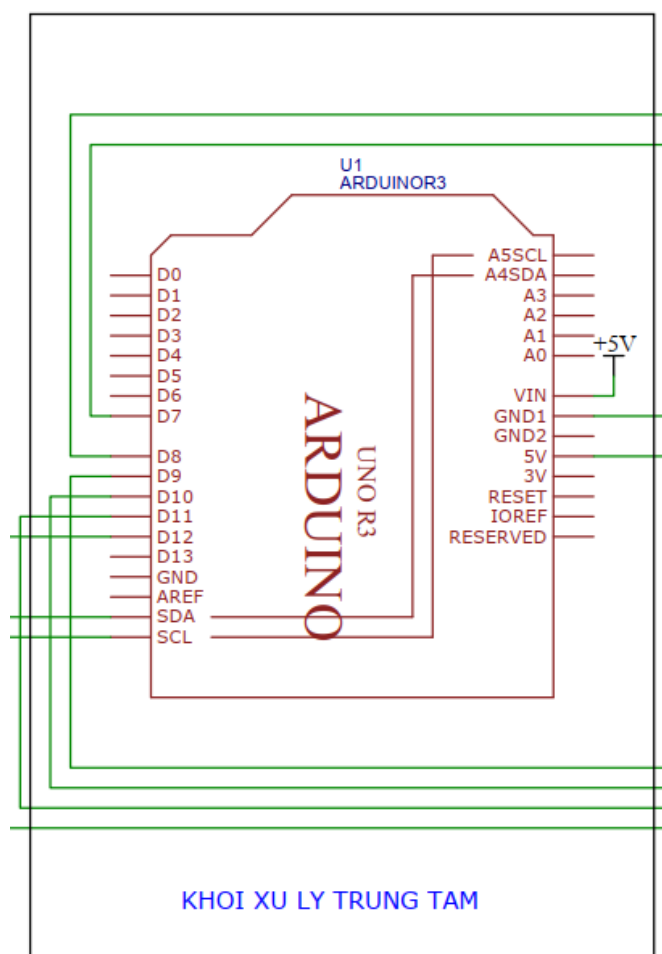
Các chân số 4 đến 7 của PCF8574 nối với các chân từ D4 đến D7 của LCD, các chân từ 0 đến 2 nối lần lượt với các chân RS, RW, E của LCD. Chân SCL và SDA kết nối trực tiếp với chân SCL và SDA trên Arduino Uno R3. Điện áp hoạt động của I2C từ 2.5-6V nên sẽ được kết nối với nguồn 5V trên chân 5V của Arduino Uno R3.

LCD1602 sẽ hiển thị kích thước của hộp gồm chiều dài, chiều rộng và chiều cao theo dạng “dàixrộngxcao” (ví dụ: 05x04x03) ở hàng 1 của LCD.

3.4.3. Khối xử lý trung tâm

Người thực hiện sẽ chọn khối xử lý trung tâm là một Arduino Uno R3 vì những ưu điểm sau:

- Giá thành hợp lý.
- Dễ dàng viết chương trình với IDE được cung cấp sẵn với nhiều thư viện tiện ích dễ dàng giao tiếp với ngoại vi.
- Số lượng chân GPIO tương đối nhiều để kết nối với các khối khác trong hệ thống.



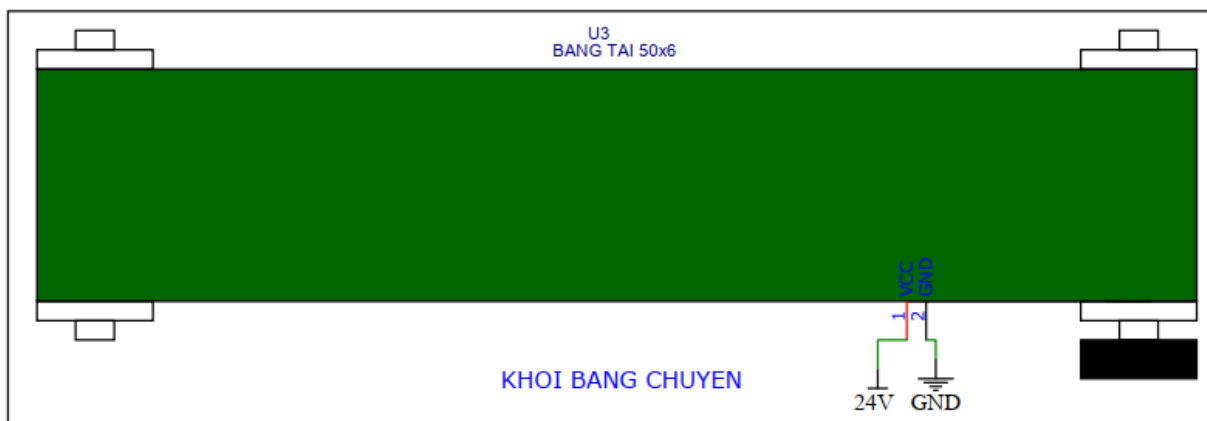
Hình 3.4 Khối xử lý trung tâm

Khối xử lý trung tâm này được cấp nguồn bởi khối nguồn 5V DC thông qua jack cắm màu đen hoặc có thể cấp nguồn từ máy tính thông qua cổng Serial bên cạnh. Chân SDA và SCL dùng để điều khiển khối hiển thị, các chân từ 7 đến 12 dùng để điều khiển và nhận tín hiệu từ khối cảm biến, các chân GND và 5V dùng để cấp nguồn cho các cảm biến trong khối cảm biến.

3.4.4. Khối băng chuyền

Khối băng chuyền người thực hiện sẽ sử dụng băng chuyền mini loại 50x6 cm để thực hiện hệ thống này bởi những ưu điểm sau:

- Nhỏ gọn phù hợp với những hệ thống đơn giản.
- Dễ dàng lắp ráp.



Hình 3.5 Khối băng chuyền

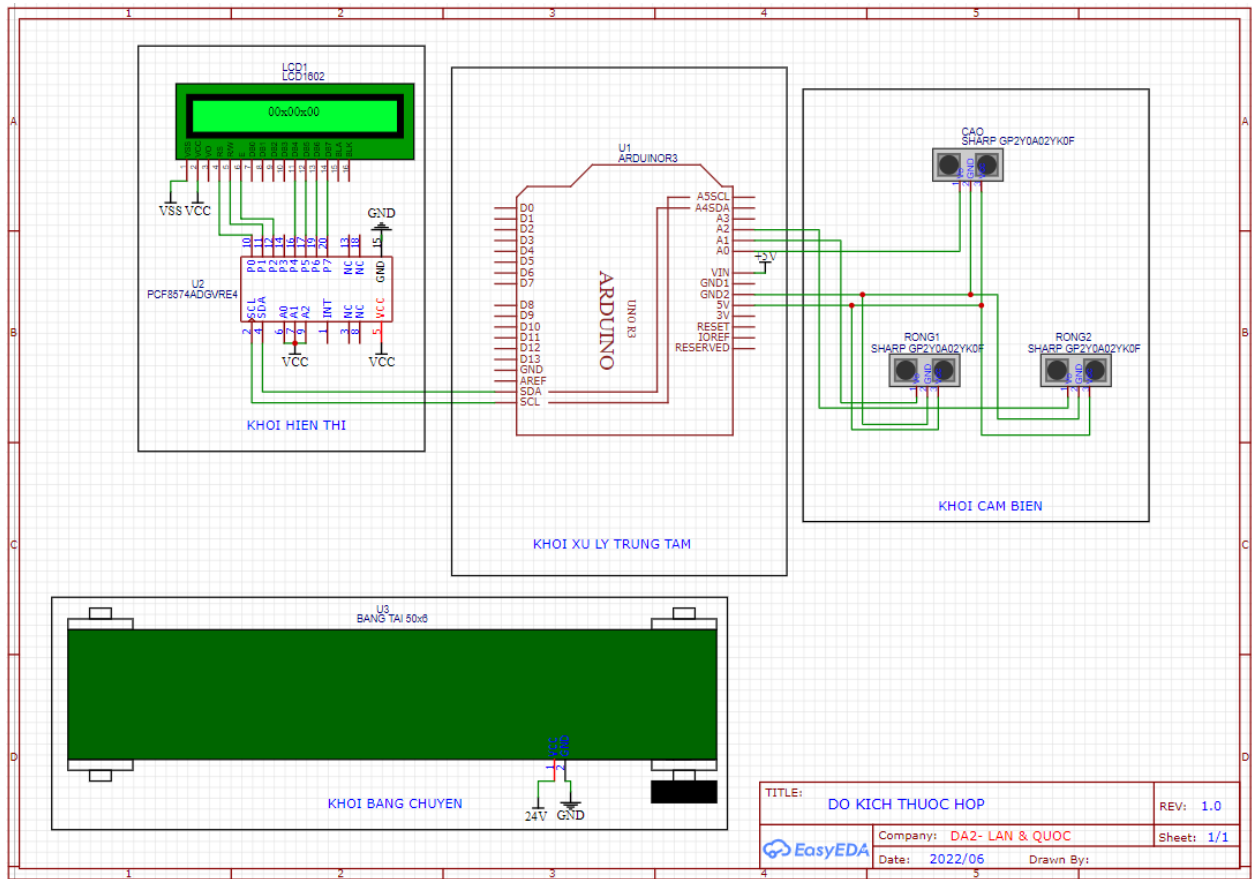
3.4.5. Khối nguồn

Ở khối nguồn người thực hiện sẽ sử dụng hai khối nguồn khác nhau là:

- Adapter 5V để cấp cho khối xử lý trung tâm, khối hiển thị và khối cảm biến vì những ưu điểm sau:
 - + Giá thành rẻ.
 - + Có thể chuyển đổi trực tiếp từ 220V AC sang 5V DC.
 - + Nhỏ gọn, dễ dàng lắp ráp và sử dụng.
 - + Không tốn thời gian thi công thiết kế.
- Nguồn tổ ong 24V-5A dùng để cấp nguồn cho khối băng chuyền hoạt động. Nguồn tổ ong này thì có những ưu điểm sau:
 - + Dễ lắp ráp.
 - + Có thể chuyển đổi trực tiếp từ 220V AC sang 24V DC.
 - + Không tốn thời gian thi công thiết kế.

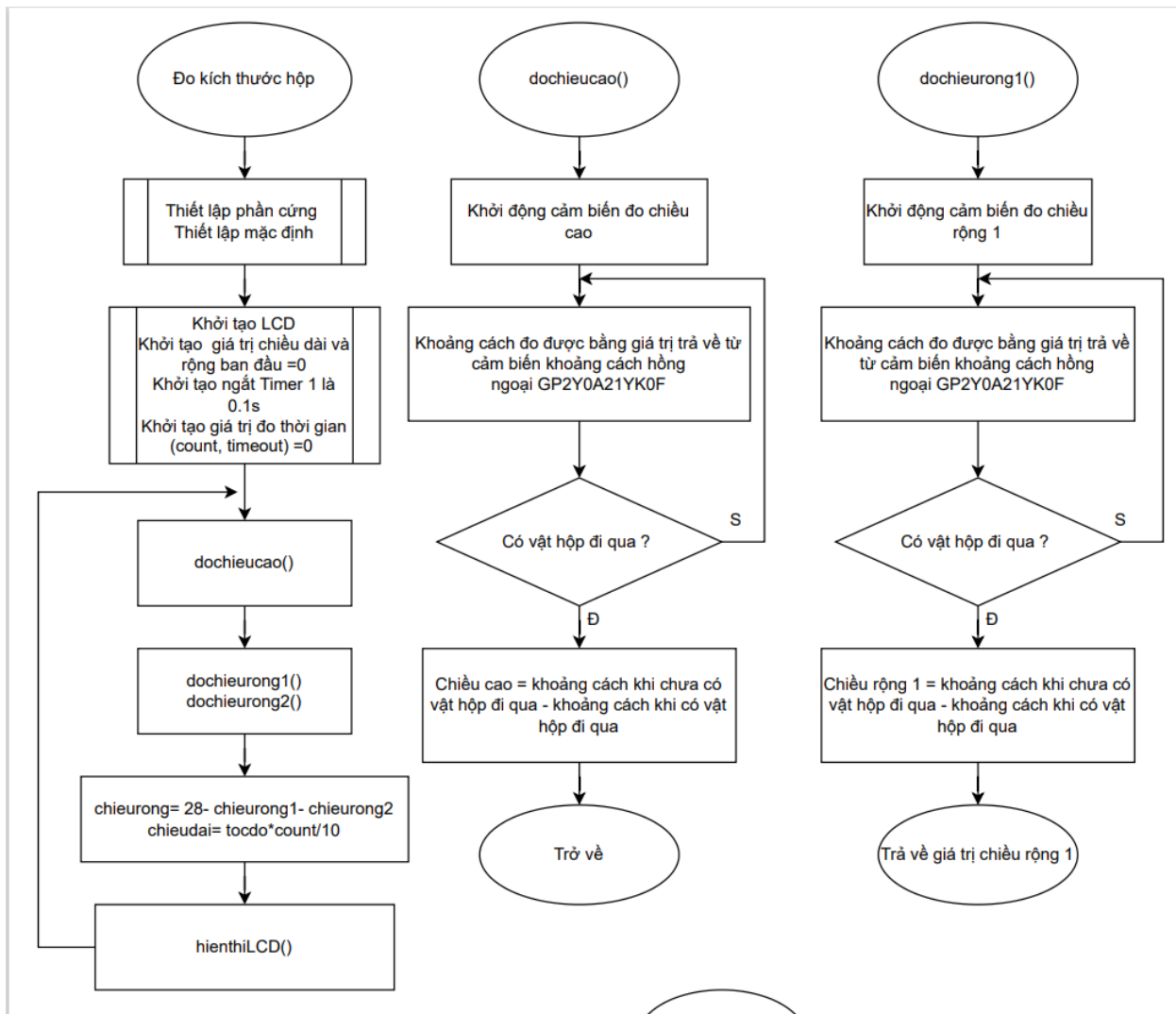
3.4.5. Sơ đồ nguyên lý toàn hệ thống

Sau khi ghép tất cả các khối chức năng lại với nhau ta được sơ đồ nguyên lý toàn hệ thống như hình 3.6:

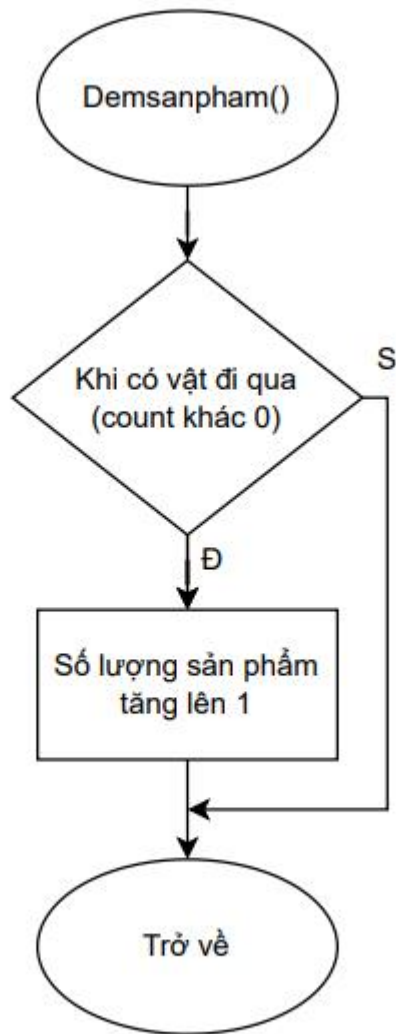


Hình 3.6 Sơ đồ nguyên lý toàn hệ thống

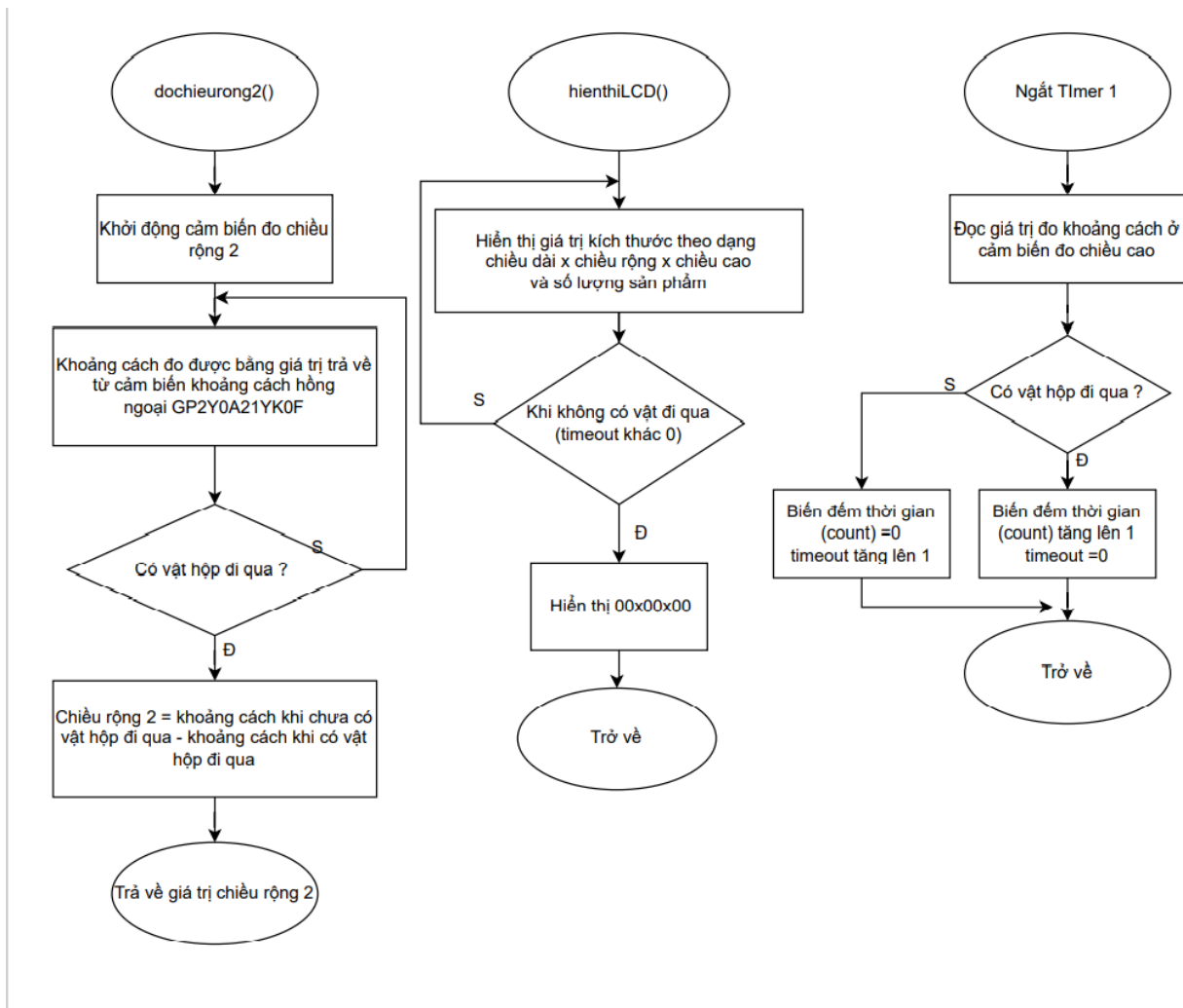
3.4.6. Lưu đồ giải thuật



Hình 3.7 Lưu đồ giải thuật phần 1



Hình 3.8 Lưu đồ giải thuật phần 2



Hình 3.9 Lưu đồ giải thuật phần 3

Giải thích:

- Ở chương trình chính, đầu tiên chúng ta sẽ thiết lập các thông số mặc định của phần cứng như các chân kết nối cảm biến cũng như LCD với khối xử lý trung tâm (ví dụ: cảm biến dùng để đo chiều cao có chân Vo nối với chân A0 của Arduino Uno R3). Tiếp theo, chúng ta thiết lập các hàm dùng để khởi động LCD1602 thông qua thư viện có sẵn trong IDE. Khởi tạo các giá trị kích thước của vật hộp ban đầu là 0, khởi tạo ngắt timer 1 là 0.1s có nghĩa là khi sự kiện khoảng cách đo được ở cảm biến đo chiều cao thay đổi thì chương trình ngắt cứ 0.1s sẽ hoạt động một lần, khởi tạo giá trị đếm thời gian count là 0 dùng để đo thời gian thay đổi của chiều cao và timeout để kiểm tra xem có vật đi qua không. Ở vòng lặp vô hạn, chúng ta sẽ tiến hành đọc các giá trị nhận về từ khối cảm biến thông qua ba chương trình con là: đo chiều cao, đo chiều rộng 1 và đo chiều rộng 2. Sau đó xử lý tính toán để cho ra được kích thước của vật hộp là chiều dài, chiều rộng, chiều cao cũng như

đếm số lượng sản phẩm và từ đó hiển thị lên màn hình LCD1602 và quá trình này lặp đi lặp lại vô hạn lần.

- Ở chương trình con đo chiều cao, đầu tiên chúng ta sẽ khởi động cảm biến khoảng cách hồng ngoại SHARP GP2Y0A02YK0F dựa trên datasheet mà nhà sản xuất cung cấp. Tiếp theo, tiến hành đọc giá trị khoảng cách đo được khi chưa có vật đi qua và đây là giá trị khoảng cách ban đầu. Nếu có vật đi qua thì khoảng cách này sẽ thay đổi không còn là giá trị ban đầu nữa thì khi đó chiều cao của vật hộp được xác định bằng cách lấy khoảng cách ban đầu đó trừ đi khoảng cách bị thay đổi khi có vật đi qua.

- Ở chương trình con đo chiều rộng 1 và đo chiều rộng 2, thì cũng tương tự như chương trình con đo chiều cao. Sau khi có giá trị chiều rộng 1 và chiều rộng 2 thì sẽ trả về hai giá trị này.

- Ở chương trình con đo chiều cao không có giá trị trả về vì khi giá trị chiều cao này được trả về thì giá trị này không thể thay đổi sau khi vật hộp đi qua khỏi cảm biến đo chiều cao có nghĩa là nó không thể trở về giá trị đo được ban đầu nên không thể đo được khoảng thời gian thay đổi của chiều cao.

- Ở chương trình ngắt timer 1 dùng để đo thời gian thay đổi khoảng cách ở cảm biến đo chiều cao lúc có vật hộp đi qua thông qua biến đếm count, khi vật đi qua thì khoảng cách sẽ thay đổi và chương trình ngắt sẽ hoạt động biến count sẽ tăng lên 1 và giá trị 1 này có đơn vị là 0.1s nên khi tính toán giá trị chiều dài ở chương trình chính chúng ta sẽ lấy giá trị count này chia cho 10 để ra được giá trị thời gian theo giây. Cùng với đó ngắt timer1 cũng được sử dụng để kiểm tra xem có vật đi qua không thông qua biến đếm timeout, nếu có vật đi qua thì timeout bằng 0 và khi không có vật đi qua thì timeout sẽ tăng lên 1.

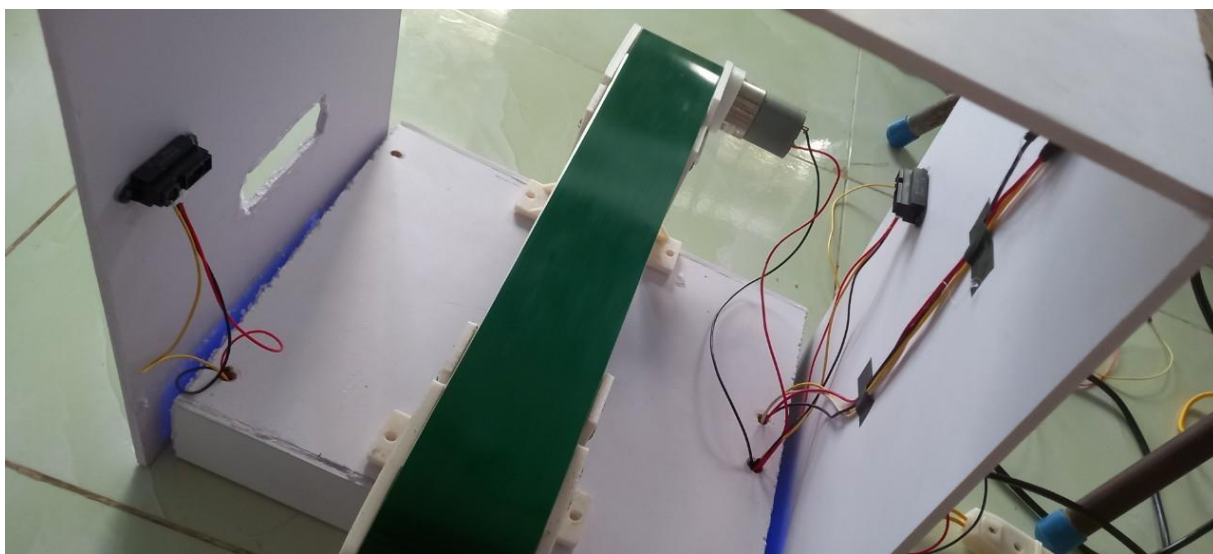
- Ở tính chiều dài “chieudai=tocdo*count/10”, “tocdo” ở đây là tốc độ băng chuyền bằng và tocdo= $(48/60)*3.14*3$, ở đây 48 là tốc độ quay số vòng/ phút của băng chuyền chia cho 60 để chuyển qua số vòng/ giây, và để chuyển từ đơn vị vòng/ giây sang cm/s chúng ta sẽ tiến hành nhân thêm với chu vi của trục quay của băng chuyền ở đây là 3 cm.

- Ở chương trình con hiển thị lên LCD thì sẽ hiển thị các giá trị kích thước theo dạng chiều dài x chiều rộng x chiều cao (ví dụ: 05x03x04) và hiển thị số lượng sản phẩm. Giá trị của timeout được sử dụng khi timeout lớn hơn một giá trị cho trước có nghĩa là giá trị này chính là khoảng thời gian bao lâu chưa có vật đi qua thì LCD sẽ hiển thị trở lại giá trị 00x00x00.

Chương 4 KẾT QUẢ THỰC HIỆN

4.1 Phần cứng của hệ thống sau khi thiết kế

Sau khi lựa chọn các linh kiện và lắp ráp theo sơ đồ nguyên lý của hệ thống ở hình 3.6 thì ta được hệ thống phần cứng như hình 4.1:

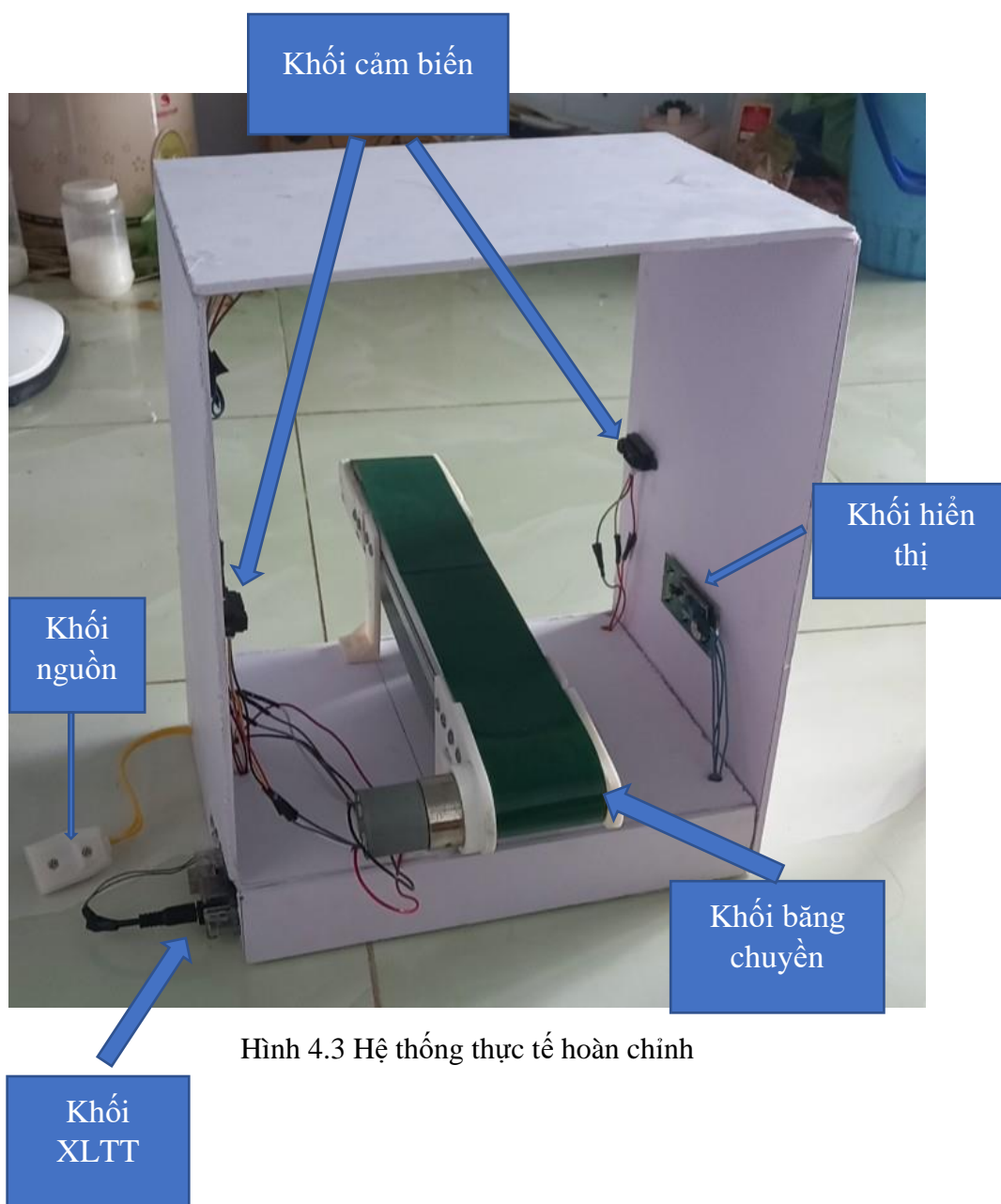


Hình 4.1 Kết nối các linh kiện phần cứng theo sơ đồ nguyên lý của hệ thống



Hình 4.2 Kết nối các linh kiện phần cứng theo sơ đồ nguyên lý của hệ thống

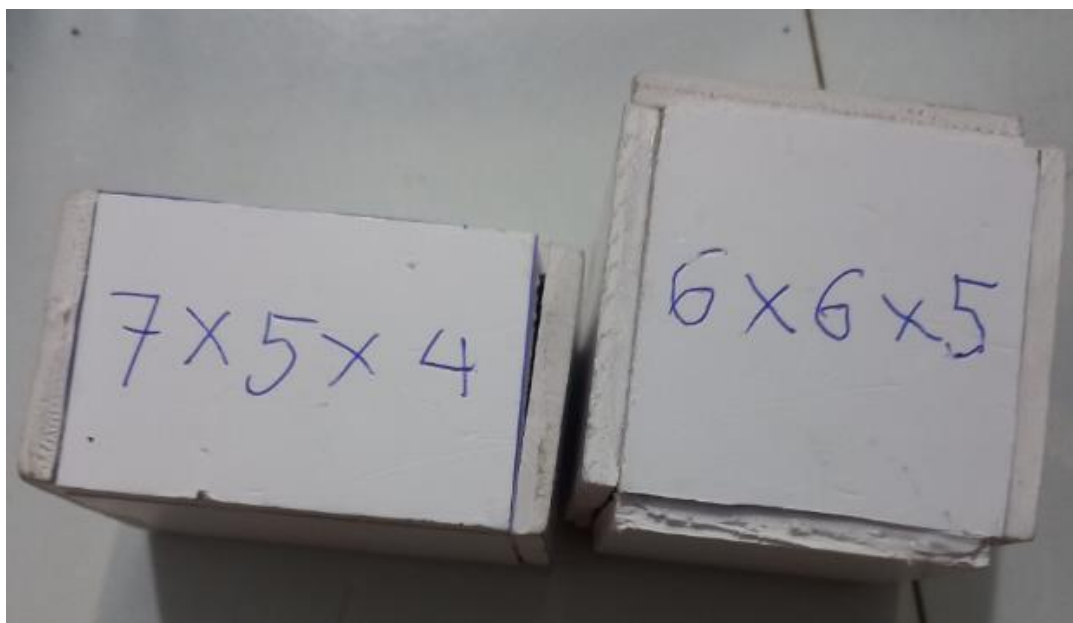
Sau khi đóng gói hết tất cả các phần của hệ thống ta được hệ thống thực tế hoàn chỉnh như hình 4.3 bên dưới:



Hình 4.3 Hệ thống thực tế hoàn chỉnh

4.2 Vật hộp mẫu để đánh giá hoạt động hệ thống

Ở đây chúng em sẽ sử dụng hai vật hộp có hai kích thước khác nhau để khảo sát hệ thống là 6x6x5 và 7x5x4:




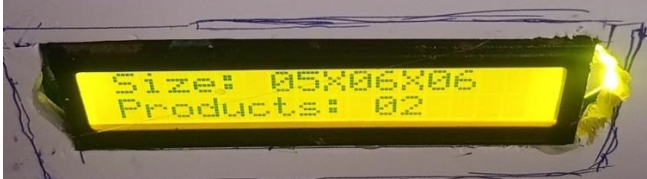
Hình 4.4 Vật hộp mẫu dùng để khảo sát


4.3 Khi hộp mẫu để thẳng hàng với băng tải

Đặt hộp lên khối băng chuyên cho hộp duy chuyển qua khu vực của khối cảm biến. Từ đó chúng ta có bảng khảo sát kết quả với hai hộp như sau:

- Với vật hộp 6x6x5:

Ta có bảng kết quả khảo sát sau:

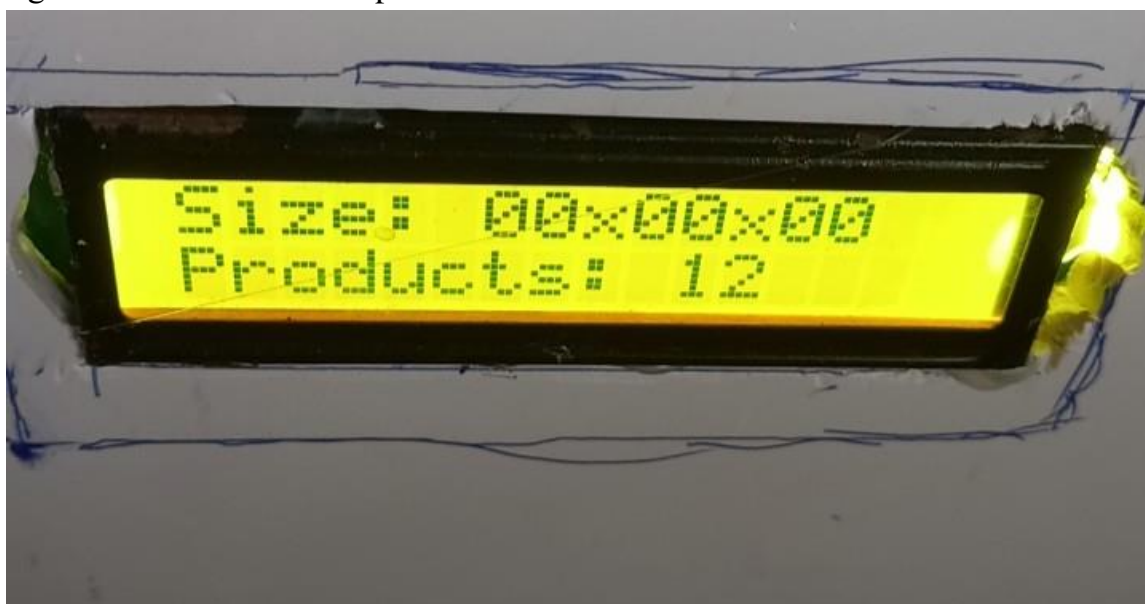
Lần đo	Hình ảnh	Kết quả
1		7x6x5
2		5x6x6

3		6x6x5
4		6x7x5
5		7x6x5

Bảng 4.1 Kết quả khảo sát hộp 6x6x5

Nhìn vào bảng khảo sát ta thấy LCD đã hiển thị được kích thước của hộp 6x6x5 qua 5 lần đo và giá trị của số lượng sản phẩm cũng đã thay đổi và hiển thị lên LCD.






Sử dụng kết quả đo lần 3 ta khảo sát được sau một thời gian không có vật hộp nào đi qua hệ thống thì giá trị kích thước sẽ reset về giá trị 00x00x00 và số lượng sản phẩm không đổi so với lần đo thứ 3 phía trên



Hình 4.5 Giá trị kích thước reset khi không có hộp đi qua hệ thống

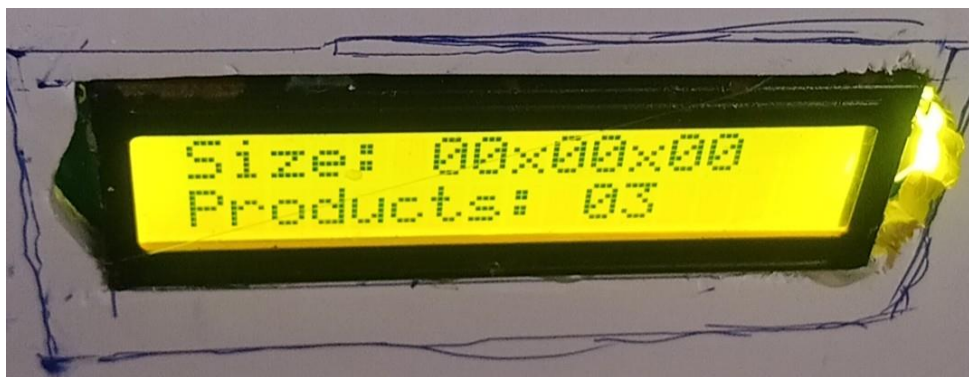
- Với vật hộp 7x6x4:

Ta có bảng kết quả khảo sát sau:

Lần đo	Hình ảnh	Kết quả
1		7x5x4
2		7x5x4
3		6x5x4
4		6x5x4
5		7x6x4

Bảng 4.2 Kết quả khảo sát hộp 7x5x4

Tương tự với hộp 6x6x5 hộp 7x5x4 cũng đã hiển thị được kích thước của hộp trên LCD và giá trị số lượng sản phẩm cũng đã thay đổi và giá trị của kích thước cũng đã reset về 00x00x00 trong một thời gian không có hộp đi qua và giá trị số lượng sản phẩm không đổi so với lần đo thứ 1 đã đề cập phía trên.



Hình 4.6 Giá trị kích thước reset khi không có hộp đi qua hệ thống

4.4 Khi hộp mẫu để nghiêng so với băng tải

- Với vật hộp 6x6x5:




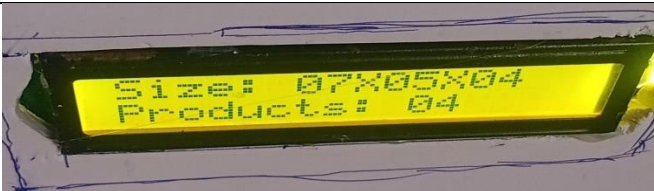

Ta có bảng kết quả khảo sát sau:

Lần đo	Hình ảnh	Kết quả
1		7x6x5
2		7x6x53
3		6x6x5
4		6x6x4
5		6x5x5

Bảng 4.3 Kết quả khảo sát hộp 6x6x5 đặt nghiêng

- Với vật hộp 7x5x4

Ta có bảng kết quả khảo sát sau:

Lần đo	Hình ảnh	Kết quả
1		6x5x4
2		7x6x4
3		6x5x4
4		7x5x4
5		7x5x4

Bảng 4.1 Kết quả khảo sát hộp 7x5x4 đặt nghiêng

Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau khi hoàn thành và thử nghiệm hệ thống thì hệ thống đã đạt được những yêu cầu sau:

- Đo được kích thước của một vật hình hộp theo giới hạn của mặt băng tải và kích thước hệ thống.
- Hiện thị được kích thước của vật hộp sau khi đo lên LCD.
- Đếm và hiện thị được số lượng sản phẩm lên LCD.

Tuy nhiên, hệ thống vẫn còn những hạn chế sau:

- Khi đặt hộp nghiêng thì sai số so với kích thước lý thuyết cao hơn so với khi để hộp thẳng.
- Do việc sử dụng cảm biến giá thành rẻ nên sai số còn nhiều.
- Không thể đo chính xác trong một lần duy nhất.
- Người thực hiện trình bày thiết kế chưa được đẹp mắt.

5.2. Hướng phát triển

Đề tài “đo kích thước vật thể hình hộp” có những hướng phát triển sau này như sau:

- Khắc phục tình trạng sai số của cảm biến bằng cách thay bằng một loại cảm biến mới.
- Tăng độ chính xác khi đặt hộp nghiêng.
- Đo chính xác trong một lần duy nhất.
- Phát triển thêm phần mềm kiểm soát số lượng cũng như kích thước từng hộp trong từng ngày.

TÀI LIỆU THAM KHẢO

- [1] Makerguides, “How to use a SHARP GP2Y0A21YK0F IR Distance Sensor with Arduino”, link: <https://www.makerguides.com/sharp-gp2y0a21yk0f-ir-distance-sensor-arduino-tutorial/>
- [2] Nshop, “Cảm Biến Khoảng Cách Hồng Ngoại Analog SHARP GP2Y0A21YK0F”, link: <https://nshopvn.com/product/cam-bien-khoang-cach-hong-ngoai-analog-sharp-gp2y0a21yk0f/>
- [3] KYSUNGHEO.COM (2021), “#1_Chuyển giao tiếp I2C”, link: <https://kysungheo.com/chuan-giao-tiep-i2c/>
- [4] Linh kiện 3M (2016), “Nguồn Tụ Ong 24V 5A (BH 6 Tháng)”, link: <https://chotroihn.vn/nguon-to-ong-24v-5a>
- [5] IOT MAKER (2022), “Arduino Uno R3”, link: <https://iotmaker.vn/arduino-uno-r3.html>
- [6] DatasheetsPDF.com (2014), “HY-SRF05 Precision Ultrasonic Sensor”, link: <https://datasheetspdf.com/pdf-file/813041/ETC/HY-SRF05/1>
- [7] DatasheetsPDF.com (2016), “8-bit AVR Microcontrollers”, link: <https://datasheetspdf.com/pdf-file/1057331/ATMEL/ATmega328/1>
- [8] Arduino.cc (2022), “Arduino® UNO R3”, link: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

PHỤ LỤC

```
#include <SharpIR.h>

#include <TimerOne.h> // khai báo thư viện timer 1

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);


boolean flag=0;

int timeout=0; //khi hiển thị LCD khi ko có vật qua thì hiện 0x0x0

int count=0;

int count_temp=0;

int pro=0;

int height;

int distance;

float v=7.5; // (48/60)*3.14*3 cm/s

/*measure width*/

int width=0;

int w1=0;

int w2=0;

int slength=0;

#define TRIG_PIN_CAO 7

#define ECHO_PIN_CAO 8


void IR_1_height()

{

    long thoigian;
```

```

digitalWrite(TRIG_PIN_CAO, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN_CAO, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN_CAO, LOW);
thoigian = pulseIn(ECHO_PIN_CAO, HIGH);
//do chieu cao
distance=(thoigian / 29.4 / 2);
if(distance>10&& distance < 18)
{
    height=20-distance;
}
}
SharpIR sensor( SharpIR::GP2Y0A21YK0F, A1 );
int IR_3_width_2()
{
    int distance_width1 = sensor.getDistance();
    if(distance_width1 < 15)
    {
        w1= distance_width1 +1;
    }
}
SharpIR sensor1( SharpIR::GP2Y0A21YK0F, A0 );
void IR_2_width_1()
{
    int distance_width2 = sensor1.getDistance();

```

```

if(distance_width2 < 15)
{
    w2= distance_width2 +1;
}
}

int count_product()
{
    if(count_temp==7 ||count_temp==6)
    {
        flag=1;
    }

    if(flag==1)
    {
        pro++;
        flag=0;
    }
}

void Display()
{
    lcd.setCursor(0,1);
    lcd.print("Products: ");
    lcd.setCursor(10,1);
    lcd.print(pro/10);
}

```

```

lcd.setCursor(11,1);
lcd.print(pro%10);
if(count!=0 && timeout<150)
{
  lcd.setCursor(0,0);
  lcd.print("Size: ");
  lcd.setCursor(6,0);
  lcd.print(slength/10);
  lcd.setCursor(7,0);
  lcd.print(slength%10);
  lcd.setCursor(8,0);
  lcd.print("X");
  lcd.setCursor(9,0);
  lcd.print(width/10);
  lcd.setCursor(10,0);
  lcd.print(width%10);
  lcd.setCursor(11,0);
  lcd.print("X");
  lcd.setCursor(12,0);
  lcd.print(height/10);
  lcd.setCursor(13,0);
  lcd.print(height%10);
}
else
{
  lcd.setCursor(0,0);

```

```

    lcd.print("Size: ");
    lcd.setCursor(6,0);
    lcd.print("00x00x00");
}
//lcd.clear();

}

void setup()
{
    Serial.begin( 9600 );
    lcd.init();
    lcd.backlight();
    pinMode(TRIG_PIN_CAO, OUTPUT);
    pinMode(ECHO_PIN_CAO, INPUT);
    Timer1.initialize(100000); // khởi tạo timer 1 đến 0.1 giây
    Timer1.attachInterrupt(IRQHandle); // khai báo ngắt timer 1
}

void loop()
{
    slength= float(v)*(count)/10;
    IR_1_height();

    IR_3_width_2();
    IR_2_width_1();

```

```

if(w2==14&&w2==13) w2=0;

width=28-w1-w2;
if(width==27) width=0;
count_product();
Serial.print(count);
Serial.print("\n");
Serial.print(timeout);
Serial.print("\n");
Display();
delay(250);
}

void IRQHandle()
{
  if( distance>10 && distance <17)
  {
    count_temp++;
    count=count_temp;
    timeout=0;
  }
  else {
    timeout++;
    count_temp=0;
    //count=0;
    noInterrupts(); } }

```