ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN - ĐIỆN TỬ

….🕮….

BÁO CÁO BÀI TẬP LỚN 2

Đo lường Công nghiệp - EE3005

ĐỀ TÀI 6:

Thiết kế mạch đo và hiển thị khoảng cách sử dụng cảm biến siêu âm

Giảng viên hướng dẫn: **Nguyễn Đức Hoàng**

Lớp: L01

Sinh viên thực hiện: **1. Nguyễn Ngọc Khanh – 2111474**

2. Trần Nguyễn Phương Linh – 2110326

TP.HCM 22/04/2024

MỤC LỤC

LỜI MỞ ĐẦU

Trong đo lường, dù với thiết bị máy móc tân tiến tới đâu thì hệ thống đo lường cũng không thể tránh khỏi sai số. Nhưng chúng ta có thể giảm thiểu độ lớn sai số của chúng bằng việc thiết kế ra một hệ thống đo lường hiệu quả hơn đồng thời phân tích và xử lý số liệu đo một cách hợp lý.

Hiệu chỉnh một thiết bị đo hay gọi tắt là Calib nghĩa là kiểm tra và điều chỉnh (nếu cần thiết) kết quả ở đầu ra của nó sao cho tương ứng với đầu vào của nó trong dải đo được quy định. Ngay cả một thiết bị quan trọng cũng sẽ trở nên vô dụng khi không được hiệu chỉnh. Trong quá trình hiệu chuẩn, điều chỉnh một bộ phận của thiết bị hay cả thiết bị là để đảm bảo nó hoạt động như mong đợi, cung cấp và truyền về kết quả chính xác, tin cậy đáp ứng các tiêu chuẩn chất lượng. Những điều chỉnh trong quá trình hiệu chuẩn phải chắc chắn sự chênh lệch nằm trong dung sai.

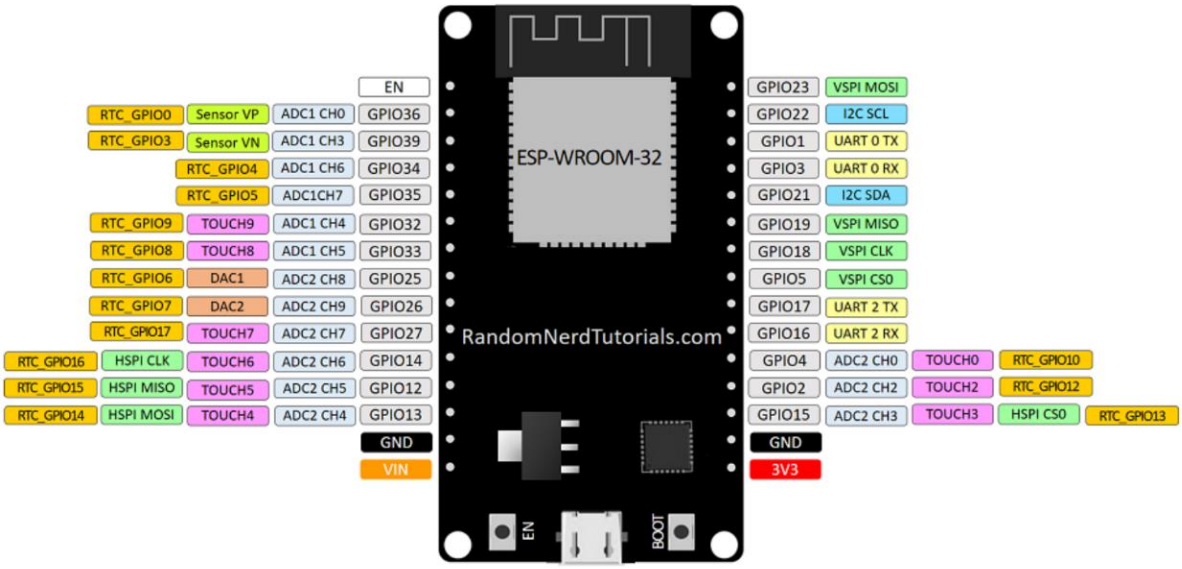
Qua đó, nhóm chúng em quyết định lựa chọn thực hiện đề tài 6 **"Thiết kế mạch đo và hiển thị khoảng cách sử dụng cảm biến siêu âm"** trong đó có ứng dụng kiến thức đã học vào việc thiết kế và Calib để mạch đo đưa ra được kết quả chuẩn xác hơn, có sai số trong khoảng cho phép. Nhóm tin rằng việc nghiên cứu và phát triển hệ thống này không chỉ mang lại hiểu biết sâu sắc về ứng dụng của việc Calib trong mạch đo, mà còn giúp chúng em áp dụng kiến thức lý thuyết vào thực tế một cách sáng tạo và hiệu quả.

Chúng em xin cảm ơn thầy Nguyễn Đức Hoàng, người đã chú ý và hỗ trợ nhóm trong quá trình nghiên cứu và phát triển đề tài này. Chúng em tin rằng sự hướng dẫn của thầy sẽ giúp đỡ chúng em không chỉ trong quá trình thực hiện đề tài mà còn trong việc xây dựng kỹ năng nghiên cứu và ứng dụng kiến thức vào thực tế. Chúng em sẽ cố gắng hết sức để đạt được những kết quả tốt nhất và hy vọng rằng đề tài này sẽ mang lại giá trị thực tiễn trong lĩnh vực đo lường trong công nghiệp. Một lần nữa, chúng em xin chân thành cảm ơn sự hỗ trợ và dạy dỗ tận tâm của thầy.

I. CƠ SỞ LÝ THUYẾT

1. Vi điều khiển ESP32

1.1. Sơ đồ chân



Hình 1. Vi điều khiển ESP32

1.2. Thông số kỹ thuật

a) CPU

- Có 32 bit.

- Tốc độ xử lý 160MHz - 240 MHz.

- Tốc độ xung nhịp đọc flash chip 40MHz - 80MHz (tùy chỉnh khi lập trình).

- ROM: 448 Kbyte ROM.

- 4MB external FLASH.

- RAM: 520 KByte SRAM, 520 KB SRAM liền chip (trong đó 8 KB RAM RTC tốc độ cao - 8 KB RAM RTC tốc độ thấp (dùng ở chế độ Deep Sleep).

b) Ngoại vi

- 18 kênh bộ chuyển đổi Analog-to-Digital (ADC), 2 × SPI, 2 × UART, 2 × I2C, 16 kênh đầu ra PWM, 2 Bộ chuyển đổi DAC, 2 × I2S, 10 GPIO cảm biến điện dung.

- Các tính năng ADC và DAC được gán cho các chân cố định. Tuy nhiên, ta có thể quyết định các chân nào là UART, 12C, SPI, PWM,… chúng ta chỉ cần khai báo trong code. Điều này có thể thực hiện được do tính năng ghép kênh của chip ESP32.

VD: Các chân từ 34 - 39 là Input Only Pins nên không thể cấu hình chúng là Ouput.

c) Ultra - Low Power

- Sleep Mode: là trạng thái ESP32 tiết kiệm năng lượng của ESP32 khi không sử dụng. Năng lượng chỉ đủ truyền cho RAM để lưu trữ dữ liệu.

- Chế độ hoạt động: Tất cả tính năng hoạt động. Dòng chip yêu cầu là 240mA, đôi khi nếu sử dụng cả Bluetooth và wifi có thể lên tới 790mA.

- Light Sleep : Tắt hết Wifi, BLE, RAM và CPU được định mức clock, dòng tiêu thụ ~ 0.8mA.

- Deep Sleep : Ở chế độ ngủ sâu, CPU, hầu hết RAM và tất cả ngoại vi bị tắt. Các phần của chip vẫn được bật là: bộ điều khiển RTC, ngoại vi RTC (bộ đồng xử lý ULP) và RTC memories. Dòng tiêu thụ 154A 0.15mA.

- Hibernate: Mọi thứ khác đều bị tắt ngoại trừ chỉ một bộ đếm thời gian RTC và một số GPIO RTC đang hoạt động. Chúng chịu trách nhiệm đánh thức chip khỏi Hibernate.

d) Wifi

- 802.11 b/g/n/e/i (Wi-Fi 2,4 GHz)

- Station mode (STA hay Wi-Fi client). ESP32 sẽ kết nối tới các điểm truy cập.

- Hoạt động như một điểm truy cập (Access Ponit mode hay Soft-AP). Nó giống như trung tâm của mọi thông tin liên lạc. Các Station sẽ kết nối tới ESP32(lúc này là Access-Point).

- AP-STA mode ESP32 sẽ đồng thời là điểm truy cập và truy cập đến điểm khác.

- Các chế độ bảo mật khác nhau cho những điều trên (WPA, WPA2, WEP…).

Lưu ý: Không thể sử dụng chân ADC2 khi sử dụng Wi-Fi.

e) Bluetooth

- Bluetooth: v4.2 BR/EDR và BLE.

- Việc hỗ trợ cả bluetooth khiến ESP32 có thể tương tác với các thiết bị như là bàn phím, chuột, điện thoại khi mà không có Wi-Fi. Bạn có thể tùy biến chức năng là BLE hay Bluetooth Classic tùy theo chức năng, tốc độ, năng lượng mà project cần đáp ứng.

2. Cảm biến siêu âm HC- SR04

2.1. Giới thiệu

Cảm biến siêu âm HC-SR04 (Ultrasonic Sensor) là thiết bị điện tử được sử dụng rất phổ biến để xác định khoảng cách từ vật thể đến cảm biến. Cảm biến có thời gian phản hồi nhanh, độ chính xác cao, phù hợp cho các ứng dụng phát hiện vật cản, đo khoảng cách bằng sóng siêu âm.

Hình 2. Cảm biến siêu âm HC-SR04

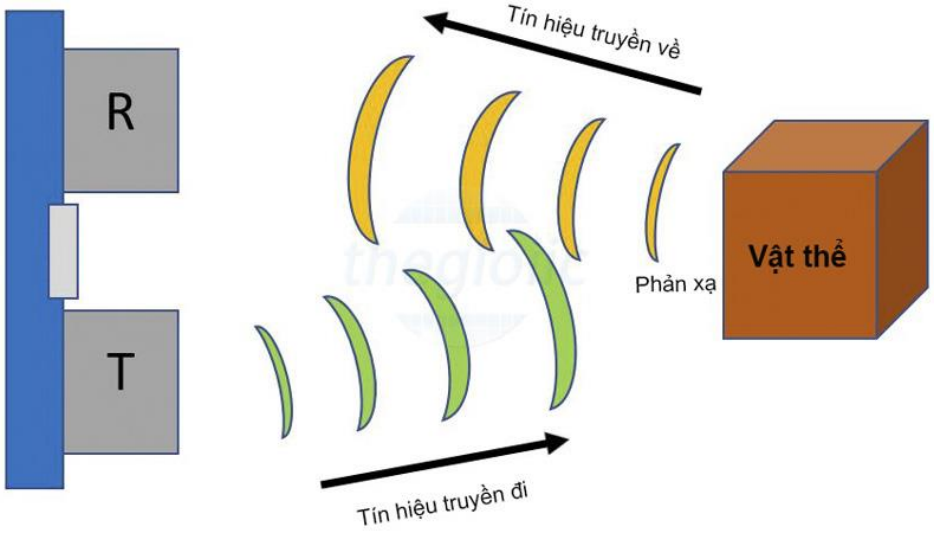
2.2. Thông số kỹ thuật

- Điện áp hoạt động: 5VDC
- Dòng tiêu thụ: 10~40mA
- Tín hiệu giao tiếp: TTL
- Chân tín hiệu: Echo, Trigger.
- Góc quét:<15 độ
- Tần số phát sóng: 40Khz
- Khoảng cách đo được: 2~450cm (khoảng cách xa nhất đạt được ở điều khiện lý tưởng với không gian trống và bề mặt vật thể bằng phẳng, trong điều kiện bình thường cảm biến cho kết quả chính xác nhất ở khoảng cách <100cm).
- Sai số: 0.3cm (khoảng cách càng gần, bề mặt vật thể càng phẳng sai số càng nhỏ).
- Kích thước: 43mm x 20mm x 17mm

2.3. Chức năng các chân

- VCC: nối nguồn 3.3V - 5V
- Trig: Là chân Digital output (điều khiển tín hiệu phát)
- Echo: là chân Digital Input (chân nhận tín hiệu phản hồi)
- GND: nối GND

2.4. Nguyên lý hoạt động



Hình 3. Nguyên lý hoạt động của cảm biến siêu âm

Nguyên lý hoạt động của cảm biến sóng siêu âm khá đơn giản. hoạt động dựa trên chu trình cho và nhận tín hiệu. Cảm biến siêu âm SR04 gồm 2 module:1 module phát ra sóng siêu âm và 1 module thu sóng siêu âm phản xạ về. Đầu tiên cảm biến sẽ phát ra 1 sóng siêu âm trong phạm vi giám sát với tần số 40kHz nếu có vật thể đi vào khu vực này, sẽ khiến sóng phản xạ trở lại lại và tác động lên module nhận sóng. dựa trên thời gian từ lúc phát đến lúc nhận sóng ta sẽ tính được khoảng cách từ cảm biến đến chướng ngại vật:

Khoảng cách = thời gian * vận tốc âm thanh (340 m/s) / 2

Lưu ý: cảm biến siêu âm có sóng phát ra theo hình nón nên ở khoảng cách xa hoặc bề mặt vật thể không bằng phẳng sẽ dễ gây nhiễu cảm biến dẫn đến khoảng cách bị sai lệch.

2.5. Ứng dụng

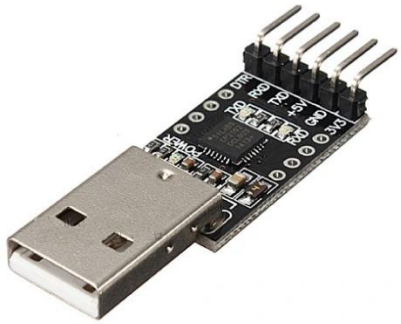
- Đếm số lượng sản phẩm
- Phát hiện chống trộm, phát hiện có người
- Tự động bật, tắt đèn hành lang

- Thước đo điện tử

3. USB UART

3.1. Giới thiệu

UART - Universal synchronous asynchronous receiver transmitter là một ngoại vi cơ bản và thường dùng trong các quá trình giao tiếp với các module như: Xbee, Wifi, Blutooth…. Khi giao tiếp UART kết hợp với các IC giao tiếp như MAX232CP, SP485EEN…. thì sẽ tạo thành các chuẩn giao tiếp RS232, RS485. Đây là các chuẩn giao tiếp thông dụng và phổ biến trong công nghiệp từ trước đến nay.



Hình 4. Mạch chuyển USB UART CP2102

3.2. Nguyên lý hoạt động

Khi ta sử dụng chân UART_CLK thì giao tiếp UART sẽ trở thành giao tiếp đồng bộ và không dùng sẽ là chuẩn giao tiếp không đồng bộ. Các bạn để ý là với bất cứ 1 chuẩn truyền thông nào, khi có sử dụng 1 chân tín hiệu làm chân CLK thì chuẩn giao tiếp đó sẽ là chuẩn giao tiếp đồng bộ và ngược lại. Ở đây mình chỉ đề cập đến giao tiếp UART không đồng bộ.

Ưu điểm của giao tiếp UART không đồng bộ: tiết kiệm chân vi điều khiển (2 chân), là ngoại vi mà bất kì 1 vi điều khiển nào cũng có, có khá nhiều module, cảm biến dùng UART để truyền nhận data với vi điều khiển. Nhược điểm của loại ngoại vi này là tốc độ khá chậm, tốc độ tối đa tùy thuộc vào từng dòng; quá trình truyền nhận dễ xảy ra lỗi nên

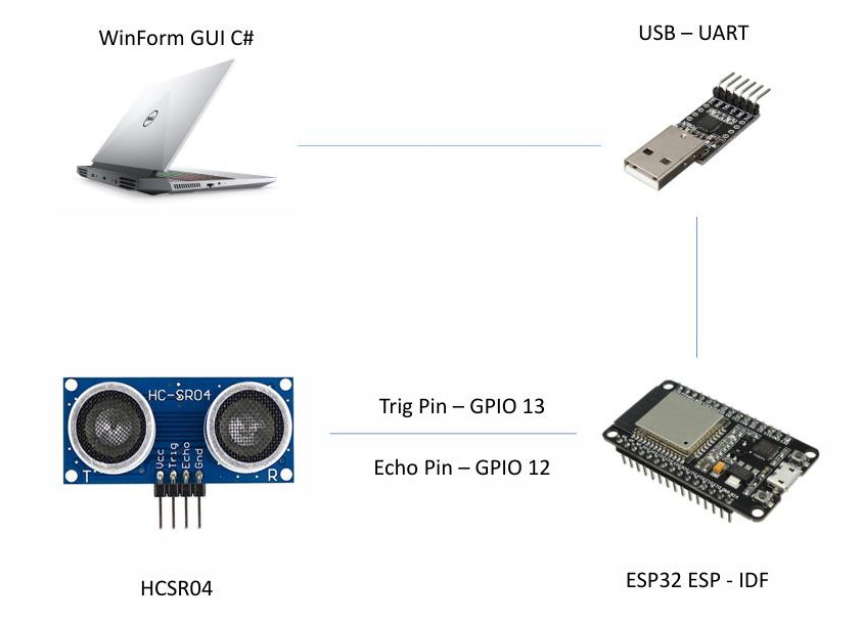
trong quá trình truyền nhận cần có các phương pháp để kiểm tra (thông thường là truyền thêm bit hoặc byte kiểm tra lỗi). UART không phải là 1 chuẩn truyền thông, Khi muốn nó là 1 chuẩn truyền thông hoặc truyền data đi xa, chúng ta cần phải sử dụng các IC thông dụng để tạo thành các chuẩn giao tiếp đáng tin cậy như RS485 hay RS232....

Thông thường chúng ta sẽ dùng ngắt nhận UART để nhận dữ liệu vì sử dụng ngắt sẽ tiện lợi, không tốn thời gian chờ cũng như mất dữ liệu. Các tốc độ thường dùng để giao tiếp với máy tính: 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200.

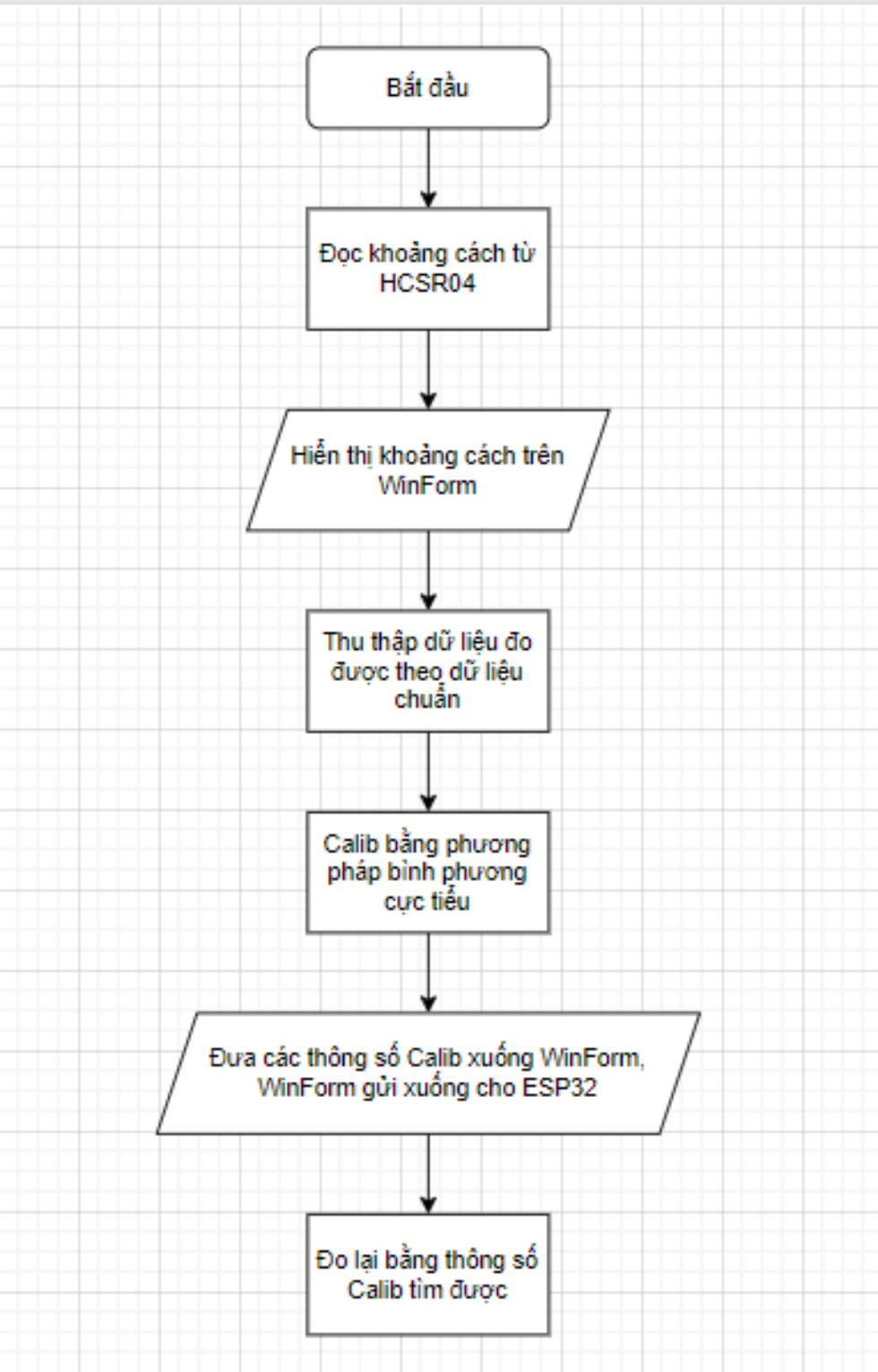
Một số phần mềm giao tiếp với máy tính: hercules_3-2-5, teraterm, Serial – Oscilloscope - v1.5...Một số modulde dùng để giao tiếp với máy tính: CP2102 USB 2.0, USB ra UART dùng PL2303, USB to UART dùng TTL FT232RL, USB ra UART dùng CH340G…

II. SƠ ĐỒ NỐI DÂY, LƯU ĐỒ GIẢI THUẬT VÀ PHƯƠNG THỨC TRAO ĐỔI DỮ LIỆU GIỮA ESP32 VỚI WINFORM GUI

1. Sơ đồ nối dây



2. Lưu đồ giải thuật



3. Phương thức trao đổi dữ liệu giữa ESP32 với WinForm GUI

- Dữ liệu nhận được từ giao diện WinForm có 2 dạng:

+ Dạng 1: "Z...C" trong đó giá trị giữa 'Z' và 'C' đại diện cho giá trị mà người dùng muốn nhập cho biến calib_Zero từ giao diện đến ESP32. Chữ 'Z' ở đầu chuỗi đại diện cho Zero, và chữ 'C' ở cuối chuỗi đại diện cho calib.

+ Dạng 2: "S...C" trong đó giá trị giữa 'S' và 'C' đại diện cho giá trị mà người dùng muốn nhập cho biến calib_SpeedofSound từ giao diện đến ESP32. Chữ 'S' ở đầu chuỗi đại diện cho SpeedofSound, và chữ 'C' ở cuối chuỗi đại diện cho calib.

+ Dựa vào chữ cái đầu tiên, vị trí của chữ cái đầu tiên và vị trí của chữ cái cuối cùng (C) trong chuỗi, chúng ta có thể xác định: biến nào giao diện hiện đang gửi dữ liệu cho và giá trị của biến đó.

- Dữ liệu truyền từ ESP32 đến giao diện có định dạng: "@...&...#", trong đó:

+ Giá trị giữa '@' và '&' đại diện cho giá trị của distance_before_calib.

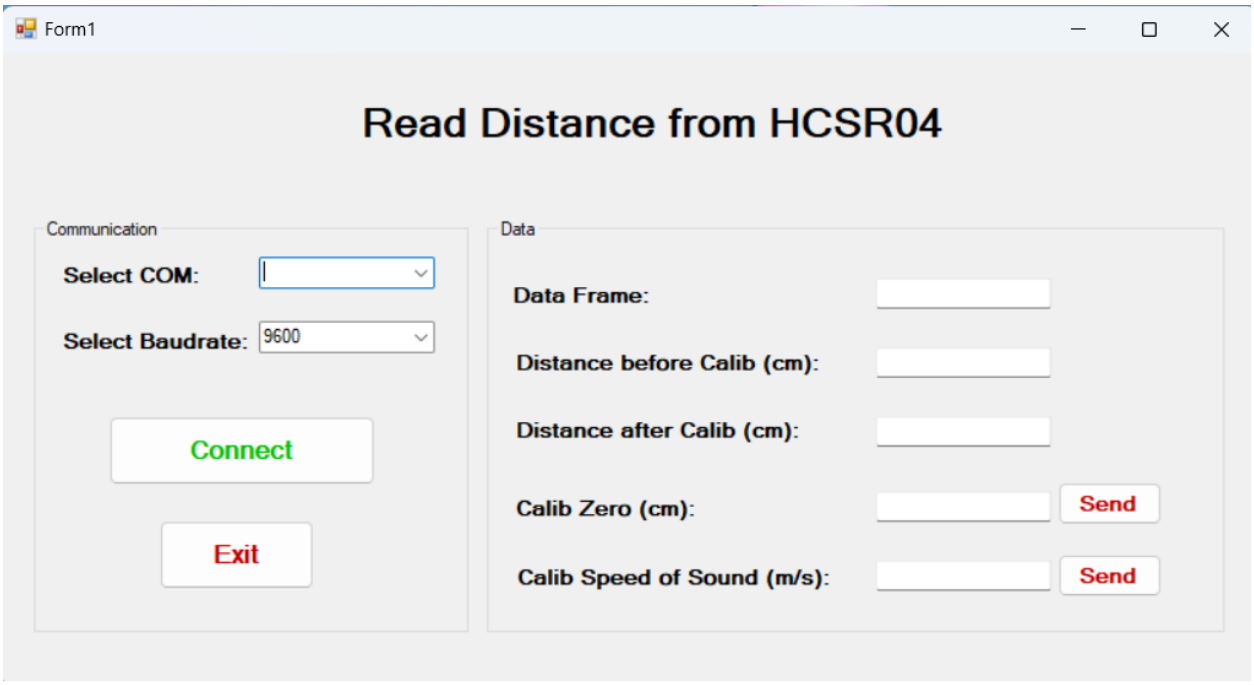
+ Giá trị giữa '&' và '#' đại diện cho giá trị của distance_after_calib. Giao diện sẽ phân tích khung này, trích xuất dữ liệu và hiển thị nó cho người dùng.

III. WINFORM GUI

Giao diện có 2 khối chính:

- Khối “Communication”: dùng để chọn cổng COM và baudrate để giao tiếp giữa GUI và ESP32

- Khối “Data”: dùng để hiển thị data frame giao tiếp giữa GUI và ESP32, cũng như khoảng cách trước khi calib, khoảng cách sau khi calib, các textbox dùng để nhập và gửi các thông số calib tìm được bằng phương pháp bình phương cực tiểu.



IV. TỔNG QUAN VỀ FRAMEWORK ESP – IDF VÀ LINK SOURCE CODE GITHUB

1. Tổng quan về ESP – IDF

Trong bài tập lớn này, em sử dụng framework ESP – IDF (Espressif IoT Development Framework) để lập trình cho ESP32 thay vì Platform IO vì:

- ESP – IDF cung cấp sự tích hợp sâu với phần cứng của Espressif, giúp tối ưu hóa hiệu suất và sử dụng tài nguyên.

- ESP – IDF cung cấp một framework chuẩn để phát triển ứng dụng trên ESP32 và ESP8266, giúp đảm bảo tính nhất quán và dễ dàng trong quá trình phát triển.

- ESP – IDF cung cấp kiểm soát cao hơn về các tính năng phần cứng của ESP32 và ESP8266, cho phép bạn tùy chỉnh và tối ưu hóa việc sử dụng các chức năng phần cứng theo nhu cầu cụ thể của dự án.

2. Link source code Github của chương trình ESP – IDF và WinForm GUI

Link source code: https://github.com/NgocKhanh0912/Calib_HCSR04.git.

V. CALIB BẰNG PHƯƠNG PHÁP BÌNH PHƯƠNG CỰC TIỂU

Ta có công thức tính khoảng cách mạch đo được như sau:

$$D = (34300 * \Delta T) / (2 * 10^6) = 0.01715 * \Delta T$$

Suy ra: $\Delta T = D / 0.01715$

Với:

+ D là khoảng cách của các lần đo (cm)

+ ΔT là chênh lệch thời gian giữa lúc phát ra tín hiệu và lúc nhận lại tín hiệu của HCSR04 (μs)

Từ công thức trên, ta có bảng thời gian phản hồi chuẩn theo khoảng cách của mạch đo khoảng cách sử dụng cảm biến siêu âm:

Thời gian ΔT chuẩn (μs)	116.62	174.93	233.24	291.55	349.85	408.16	466.47	524.78	583.09
Khoảng cách (cm)	2	3	4	5	6	7	8	9	10

Thời gian ΔT chuẩn (μs)	641.4	699.71	758.02	816.33	874.64	932.94	991.25	1049.56	1107.87	1166.18
Khoảng cách (cm)	11	12	13	14	15	16	17	18	19	20

Bảng kết quả đo thực tế của mạch đo khoảng cách sử dụng cảm biến siêu âm:

Thời gian ΔT chuẩn (μs)	116.62	174.93	233.24	291.55	349.85	408.16	466.47	524.78	583.09
Đo lần 1 (cm)	2.11	2.52	3	3.55	4.03	5.4	6.83	7.79	9.64
Đo lần 2 (cm)	2.11	2.52	3	3.55	4.03	5.4	6.83	7.79	9.64
Đo lần 3 (cm)	2.11	2.54	3	3.55	4.03	5.4	6.83	7.79	9.62
Đo lần 4 (cm)	2.11	2.54	3	3.55	4.03	5.4	6.83	7.79	9.62
Trung bình các lần đo (cm)	2.11	2.53	3	3.55	4.03	5.4	6.83	7.79	9.63

Thời gian ΔT chuẩn (μs)	641.4	699.71	758.02	816.33	874.64	932.94	991.25	1049.56	1107.87	1166.18
Đo lần 1 (cm)	10.58	11.11	12.19	13.15	13.81	14.77	15.78	16.36	17.34	18.35
Đo lần 2 (cm)	10.58	11.11	12.19	13.15	13.81	14.77	15.78	16.36	17.34	18.35
Đo lần 3 (cm)	10.58	11.13	12.19	13.15	13.81	14.77	15.78	16.36	17.34	18.35
Đo lần 4 (cm)	10.58	11.13	12.19	13.15	13.81	14.77	15.78	16.36	17.34	18.35
Trung bình các lần đo (cm)	10.58	11.12	12.19	13.15	13.81	14.77	15.78	16.36	17.34	18.35

Từ số liệu trên, ta dùng phương pháp bình phương cực tiểu để tìm phương trình đường chuẩn có dạng y = Ax + B cho khoảng cách đo được theo thời gian phản hồi:

$$\begin{cases} nA + \left(\sum_{k=1}^{n} x_k\right) B = \sum_{k=1}^{n} y_k \\ \left(\sum_{k=1}^{n} x_k\right) A + \left(\sum_{k=1}^{n} x_k^2\right) B = \sum_{k=1}^{n} x_k y_k \end{cases}$$

Với n=19, $\sum_{k=1}^{19} x_k = 12186.59$, $\sum_{k=1}^{19} y_k = 188.32$, $\sum_{k=1}^{19} x_k^2 = 9754428.587$,

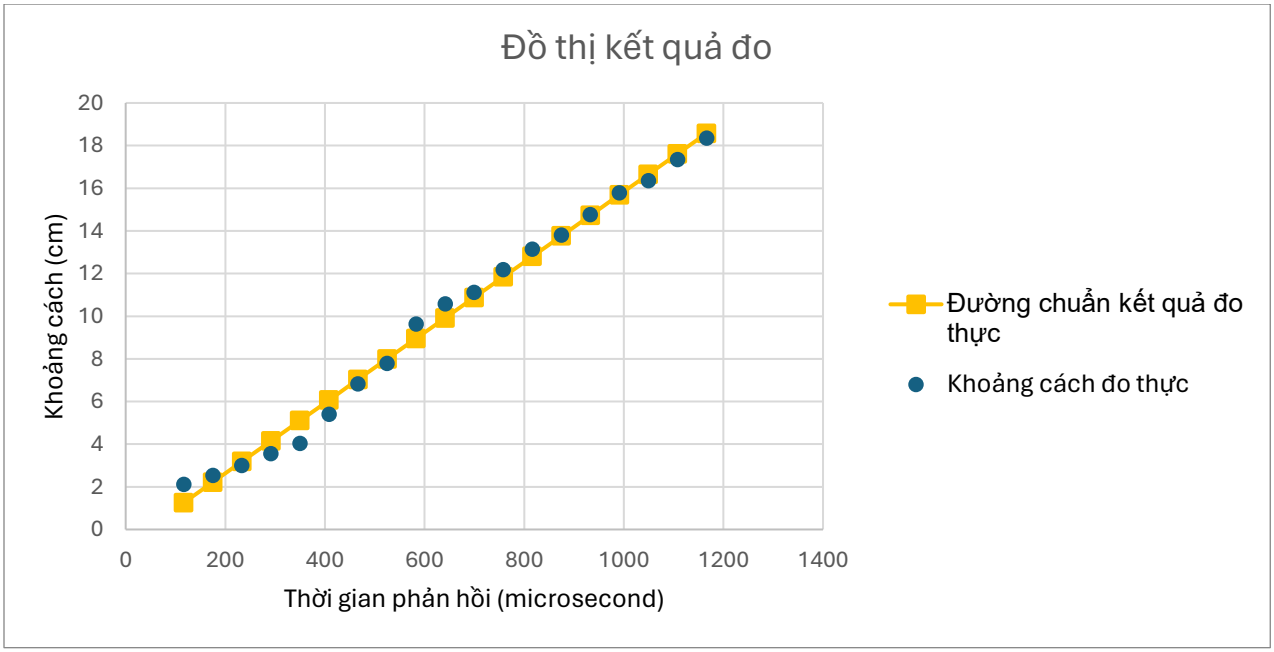
$\sum_{k=1}^{19} x_k y_k = 152749.1942$, ta có:

$$\begin{cases} 19A + 12186.59B = 188.32 \\ 12186.59A + 9754428.587B = 152749.1942 \end{cases}$$

$$\Rightarrow \begin{cases} A = -0.6664 \\ B = 0.0165 \end{cases}$$

$$\Rightarrow y = -0.6664 + 0.0165x$$

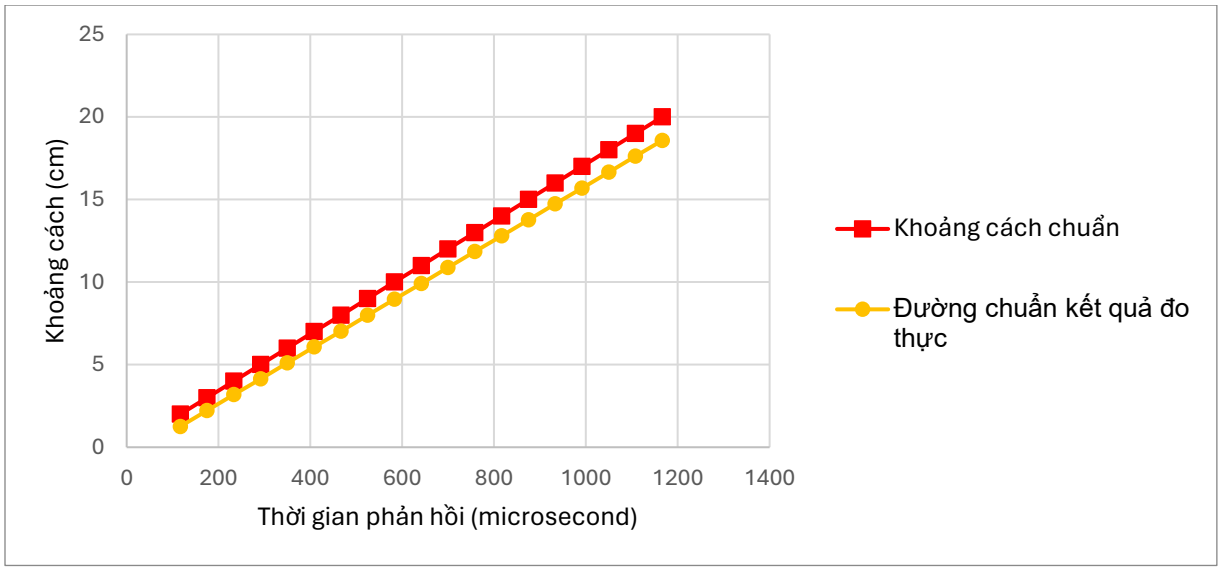
$$\Rightarrow D_{đo} = -0.6664 + 0.0165\Delta T$$



Calib số liệu:

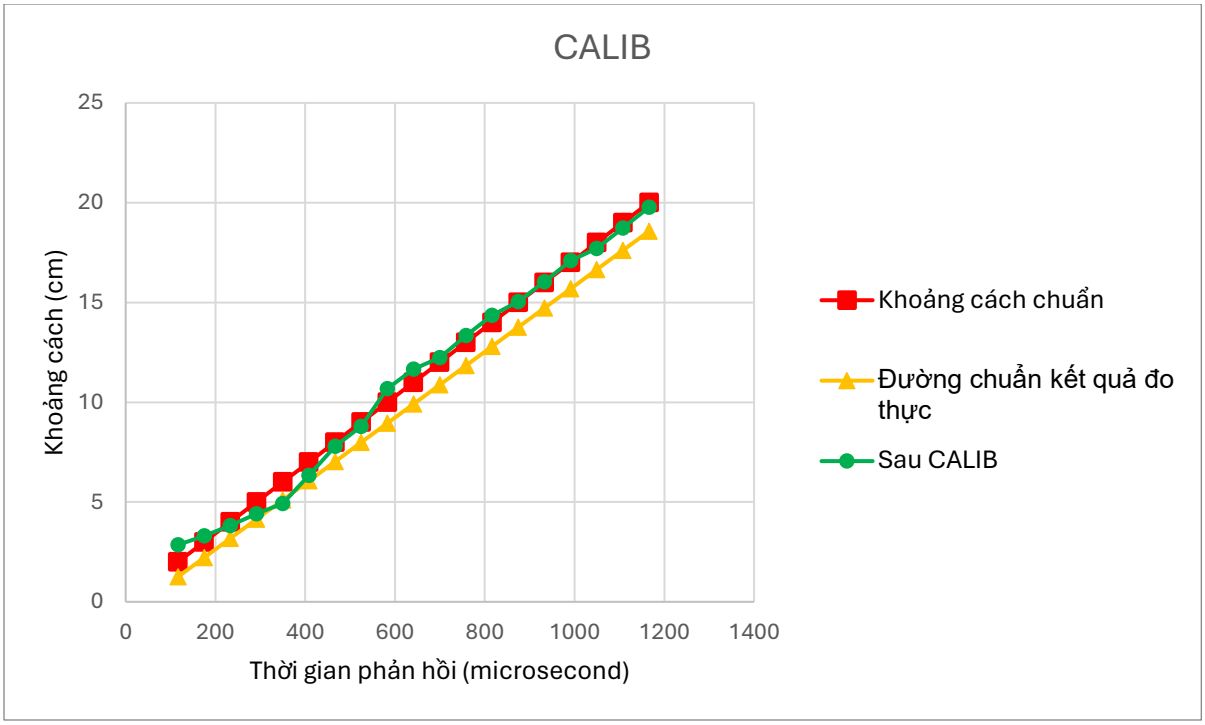
Theo công thức đo khoảng cách, ta có: $D = 0.01715 * \Delta T$

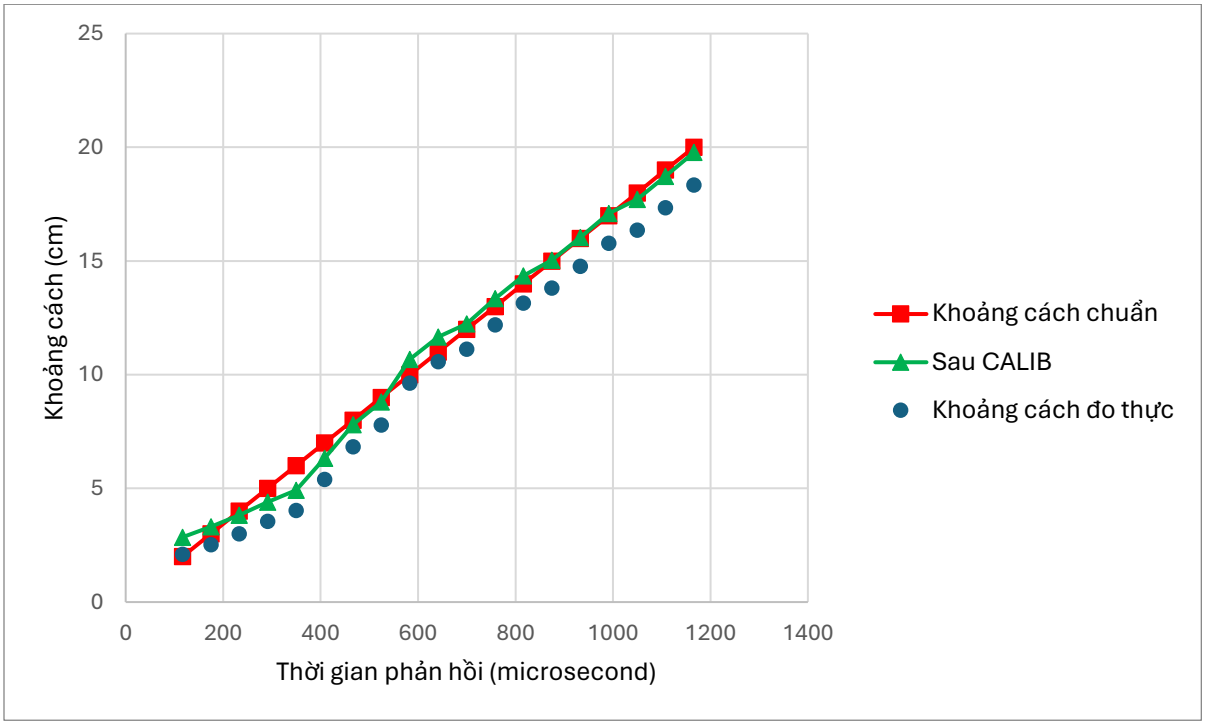
Mà thời gian đo được lại có phương trình: $D_{đo} = -0.6664 + 0.0165\Delta T$



Vậy, với mạch đo khoảng cách dùng cảm biến siêu âm như nhóm thiết kế, để thu được kết quả chính xác nhất, ta chỉnh zero = -0.6664 và span = 20 cm với độ nhạy B = 0.0165 (độ trôi độ nhạy = $0.01715 - 0.0165 = 0.00065$)

$\Rightarrow D_{CALIB} = D_{đo} + 0.6664 + 0.00065 * \Delta T$ (cm)





*** <u>Chú thích:</u>**

Khoảng cách chuẩn: Đường nối các điểm thể hiện khoảng cách chính xác cần đo được theo thời gian phản hồi chuẩn ΔT (μs)

Đường chuẩn kết quả đo thực: Từ số liệu thực đo được, ta có đường chuẩn cho khoảng cách mạch đo được trước hiệu chỉnh theo thời gian phản hồi chuẩn ΔT (μs)

Khoảng cách đo thực: Các điểm thể hiện khoảng cách mạch đo được trước hiệu chỉnh theo thời gian phản hồi chuẩn ΔT (μs)

Sau CALIB: Đường nối các điểm thể hiện khoảng cách mạch đo được sau hiệu chỉnh theo thời gian phản hồi chuẩn ΔT (μs)

VI. TỔNG KẾT

Thông qua việc thực hiện đề tài "Thiết kế mạch đo và hiển thị khoảng cách sử dụng cảm biến siêu âm", nhóm đã:

- Biết cách đọc, xử lý và tính toán khoảng cách sử dụng cảm biến siêu âm
- Biết cách calib lại khoảng cách mà cảm biến đo được dựa trên giá trị chuẩn
- Biết cách sử dụng phần mềm Visual Studio để làm giao diện WinForm cho việc calib

KẾT LUẬN

Tóm lại, trong đo lường công nghiệp, dù công nghệ hiện đại đến đâu cũng không tránh khỏi những sai số. Mạch sử dụng cảm biến siêu âm đo khoảng cách do nhóm chúng em thiết kế cũng không ngoại lệ, kết quả trả về cũng có những sai lệch nhất định. Do đó, việc hiệu chỉnh số liệu là vô cùng cần thiết.

Đề tài này không chỉ là kết quả của sự nỗ lực cá nhân mà còn là sản phẩm của sự hợp tác và trao đổi ý kiến tích cực giữa các thành viên trong nhóm. Chúng em đã cùng nhau nghiên cứu, thảo luận, và thực hiện các thí nghiệm để đảm bảo tính hiệu quả và độ chính xác của hệ thống nhúng được thiết kế.

Qua thời gian nghiên cứu thực hiện, chúng em đã phần nào nắm được căn bản nguyên tắc việc Calib dữ liệu của một mạch đo cụ thể. Bên cạnh đó, nhóm chúng em còn học hỏi được thêm cách thiết kế một mạch đo có ứng dụng thực tế cuộc sống. Nhờ đó, chúng em cũng nhận thức được tầm quan trọng của việc áp dụng Calib kết quả vào một ứng dụng đo thực tế, đặc biệt là trong một mạch đo và hiển thị khoảng cách sử dụng cảm biến siêu âm.

Chúng em một lần nữa bày tỏ lòng biết ơn chân thành đến thầy Nguyễn Đức Hoàng, người đã hướng dẫn và hỗ trợ nhóm suốt quá trình thực hiện đề tài. Sự tận tâm và kiến thức sâu rộng của thầy đã là nguồn động viên lớn, giúp chúng em hoàn thành đề tài này. Đồng thời, chúng em cam kết sẽ tiếp tục nỗ lực, phát triển thêm về lĩnh vực này và ứng dụng kiến thức đã học vào thực tế, góp phần vào sự phát triển của ngành đo lường trong công nghiệp.

Xin chân thành cảm ơn thầy!

TÀI LIỆU THAM KHẢO

[1] Cảm biến siêu âm HC-SR04. Nshopvn.com. https://nshopvn.com/product/cam-bien-sieu-am-hc-sr04/

[2] Điện tử Nhật Tùng (10/2/2022). *SRF04 Giao Tiếp STM32, Cảm biến Khoảng Cách + LCD1602 + STM*. HUỲNH NHẬT TÙNG. https://huynhnhattung.com/srf04-giao-tiep-stm32-cam-bien-khoang-cach-lcd1602-stm/

[3] Tuan Doan. *ESP32: Tổng Quan VỀ ESP32*. Deviot thời sư kĩ thuật và IOT. https://deviot.vn/tutorials/esp32.66047996/tong-quan-ve-esp32.18482631

[4] Xuan Minh. *Bài 08: Uart Trong STM32F103*. chia sẻ các vấn đề điện tử. https://laptrinharmst.blogspot.com/2018/03/bai-08-uart-trong-stm32f103.html