TRƯỜNG ĐẠI HỌC PHENIKAA
KHOA ĐIỆN – ĐIỆN TỬ

BÁO CÁO
KẾT THÚC HỌC PHẦN

ĐỀ TÀI: THIẾT KẾ CHẾ TẠO CÂN ĐIỆN TỬ ĐO KHỐI LƯỢNG

Tên học phần: Kỹ thuật vi xử lý và vi điều khiển
Học kỳ: II – Năm học: 2024 – 2025

Sinh viên thực hiện:
Nguyễn Đức Bình - Nguyễn Đức Anh
Mã sinh viên: **22010960 - 22011198**
Lớp: **EEE703046-1-2-24(N03)**
Khoa: **Điện – Điện tử**
GVHD: **TS.Vũ Quốc Đông**

HÀ NỘI, 3/2025

PHIẾU ĐÁNH GIÁ

Điểm số	Điểm chữ	Cán bộ chấm 1:
		Cán bộ chấm 2:

Rubric chấm điểm:

Tiêu chí đánh giá	Điểm 0-3.9	Điểm 4.0-5.4	Điểm 5.5-6.9	Điểm 7.0-8.4	Điểm 8.5-10	Trọng số (%)
Xác định các vấn đề cần giải quyết	Không xác định được yêu cầu bài toán	Xác định được yêu cầu bài toán	Xác định và phân tích được được yêu cầu bài toán	Xác định và phân tích được được yêu cầu bài toán. Phân tích khả năng giải quyết bài toán bằng vi đều khiển	Xác định và phân tích được được toàn bộ yêu cầu bài toán. Phân tích khả năng giải quyết bài toán bằng vi đều khiển kèm theo các ràng buộc liên quan	10
Thiết kế mạch nguyên lý và PCB	Không thiết kế được sơ đồ nguyên lý và PCB layout	Thiết kế được sơ đồ nguyên lý, không thiết kế được PCB layout	Thiết kế sơ đồ nguyên lý và PCB layout ở mức độ cơ bản	Thiết kế sơ đồ nguyên lý và có tính toán tham số các linh kiện; PCB layout thiết kế gọn gàng, đường mạch hợp lý	Thiết kế sơ đồ nguyên lý và tính toán đúng, đầy đủ tham số các linh kiện; PCB layout thiết kế gọn gàng, đường mạch hợp lý, bố trí linh kiện khoa học	30

Chất lượng sản phẩm	Sản phẩm không đáp ứng hoặc chỉ đáp ứng một phần nhỏ các yêu cầu hoặc mục tiêu của dự án. (Chỉ có mạch mô phỏng hoặc giải thuật lập trình)	Sản phẩm không đáp ứng đầy đủ các yêu cầu hoặc mục tiêu của dự án, có ít giá trị thực tế. (Có mạch mô phỏng, giải thuật, và code lập trình, mô hình chạy được)	Sản phẩm đáp ứng một số mục tiêu và yêu cầu của dự án. (Có mạch mô phỏng, giải thuật, code lập trình, mạch cắm breadboard, các mô hình mô phỏng và thử nghiệm đều chạy được)	Sản phẩm đáp ứng các mục tiêu và yêu cầu của dự án một cách đầy đủ và hiệu quả. (Có mạch mô phỏng, giải thuật, code lập trình, làm mạch in thủ công, các mô hình mô phỏng và thử nghiệm đều chạy được)	Sản phẩm đạt được tất cả các mục tiêu và yêu cầu của dự án một cách xuất sắc và có tính độc đáo, sáng tạo. (Có mạch mô phỏng, giải thuật, code lập trình, làm mạch in gọn đẹp, các mô hình mô phỏng và thử nghiệm đều chạy được)	40
Các ứng dụng, hạn chế và hướng phát triển	Không nêu được ứng dụng, hạn chế và hướng phát triển	Nêu được 1 trong 3 nội dung: ứng dụng, hạn chế và phướng phát triển của đề tài	Nêu được các ứng dụng, hạn chế và/hoặc hướng phát triển của đề tài	Phân tích đầy đủ ứng dụng, các hạn chế hoặc hướng phát triển của đề tài	Phân tích đầy đủ ứng dụng, các hạn chế và hướng phát triển của đề tài	10
Hoạt động nhóm	Không làm việc nhóm hoặc có thành viên không nắm được công việc của mình	Có kế hoạch phân chia công việc cụ thể cho các thành viên	Các thành viên tham gia thực hiện công việc của mình theo phân công	Các thành viên tham gia thực và thường xuyên trao đổi với các thành viên trong nhóm	Các thành viên tham gia và thường xuyên trao đổi, hiểu rõ công việc và nội dung của các thành viên khác trong nhóm	10

PHÂN CÔNG NHÓM

I) Thành viên nhóm

STT	Họ và Tên	Thông tin cá nhân
1	Nguyễn Đức Bình	- Mã SV: 22010960 - Lớp: EEE703046-1-2-24(N03) - Trường Đại học Phenikaa - SĐT: 0906159049 - Nơi ở: Hà Nội
2	Nguyễn Đức Anh	- Mã SV: 22011198 - Lớp: EEE703046-1-2-24(N03) - Trường Đại học Phenikaa - SĐT: 0355409517 - Nơi ở: Hà Nội

II) Phân công nhiệm vụ

Thành viên	Nhiệm vụ được giao	Nhóm đánh giá
Nguyễn Đức Bình	Mô phỏng sơ đồ nguyên lý trên Proteus, lập trình, xây dựng sơ đồ nguyên lý	**Đạt**
Nguyễn Đức Anh	Vẽ PCB bằng Kicad từ sơ đồ nguyên lý, lập trình giao diện, thiết kế vỏ gá	**Đạt**

MỤC LỤC

MỤC LỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

GIỚI THIỆU

Vi xử lý và vi điều khiển đóng vai trò quan trọng trong các hệ thống nhúng hiện đại, đặc biệt là trong lĩnh vực đo lường và điều khiển. Một trong những ứng dụng phổ biến của vi điều khiển là hệ thống cân điện tử, sử dụng cảm biến **loadcell** kết hợp với mạch khuếch đại tín hiệu **HX711** để đo khối lượng.

Cảm biến loadcell hoạt động dựa trên nguyên lý biến dạng điện trở, chuyển đổi lực tác động thành tín hiệu điện. Tuy nhiên, tín hiệu đầu ra của loadcell thường rất nhỏ và dễ bị ảnh hưởng bởi nhiễu từ môi trường. Để xử lý vấn đề này, mạch khuếch đại HX711 được sử dụng để khuếch đại và chuyển đổi tín hiệu từ loadcell thành dữ liệu số.

LÝ DO CHỌN ĐỀ TÀI

Hệ thống cân điện tử ngày càng được ứng dụng rộng rãi trong đời sống và công nghiệp, từ cân thương mại đến cân đo trong sản xuất. Tuy nhiên, độ chính xác của hệ thống này có thể bị ảnh hưởng bởi các yếu tố như nhiễu tín hiệu, dao động điện áp hoặc tác động vật lý không mong muốn.

Vì vậy, nghiên cứu và áp dụng thuật toán **Low Pass Filter (LPF)** vào quá trình xử lý tín hiệu giúp giảm nhiễu và cải thiện độ ổn định của phép đo. Điều này góp phần nâng cao hiệu suất và độ chính xác của cân điện tử, giúp hệ thống có thể hoạt động ổn định trong nhiều điều kiện môi trường khác nhau.

MỤC TIÊU NGHIÊN CỨU

- Nghiên cứu nguyên lý hoạt động của **cảm biến loadcell** và **mạch khuếch đại HX711**.
- Xây dựng hệ thống **cân điện tử** sử dụng vi điều khiển để đọc và xử lý dữ liệu từ loadcell.
- Ứng dụng thuật toán **Low Pass Filter (LPF)** để lọc nhiễu và cải thiện độ chính xác của kết quả đo.

INTRODUCTION

Microprocessors and microcontrollers play a crucial role in modern embedded systems, especially in measurement and control applications. One of the most common applications of microcontrollers is **electronic weighing systems**, which use **load cell sensors** combined with the **HX711 amplifier** to measure weight.

Load cells operate based on the strain gauge principle, converting applied force into an electrical signal. However, the output signal from a load cell is typically very weak and susceptible to environmental noise. To address this issue, the **HX711 module** is used to amplify and digitize the signal for processing.

REASON FOR CHOOSING THE TOPIC

Electronic weighing systems are widely used in both commercial and industrial applications, ranging from retail scales to manufacturing measurement systems. However, the accuracy of these systems can be affected by factors such as signal noise, voltage fluctuations, or unintended physical disturbances.

Therefore, implementing a **Low Pass Filter (LPF)** in signal processing can help reduce noise and improve measurement stability. This enhances the performance and accuracy of electronic scales, ensuring reliable operation in various environmental conditions.

RESEARCH OBJECTIVES

- Study the working principle of the **load cell sensor** and **HX711 amplifier module**.
- Develop an **electronic weighing system** using a microcontroller to acquire and process load cell data.
- Apply the **Low Pass Filter (LPF)** algorithm to filter noise and enhance measurement accuracy.

CHƯƠNG 1: TỔNG QUAN VỀ VI XỬ LÝ VI ĐIỀU KHIỂN

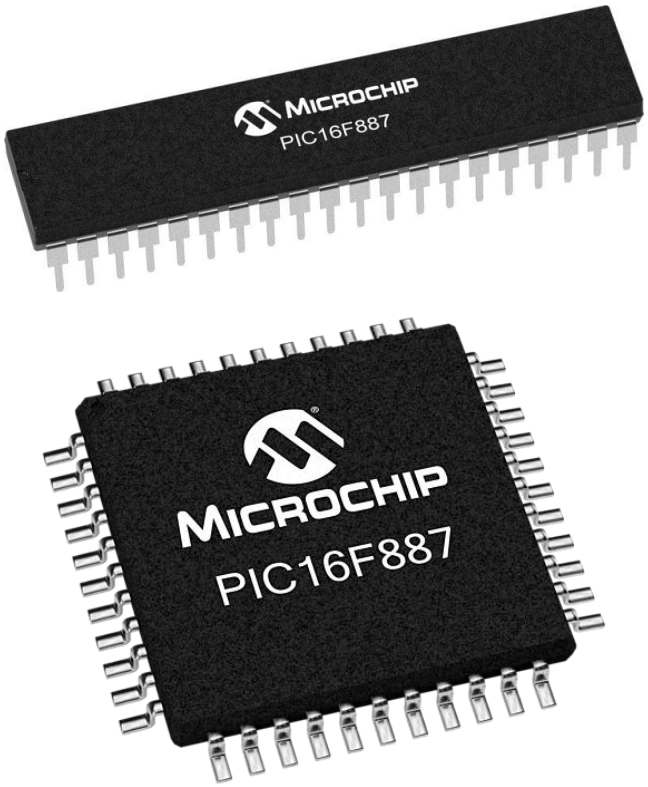
1.1. TỔNG QUAN VI ĐIỀU KHIỂN

Thông thường có 4 họ vi điều khiển 8 bit chính là 6811 của Motorola, 8051 của Intel, Z8 của Xilog và Pic 16 của Microchip Technology. Mỗi một loại trên đây đều có một tập lệnh và thanh ghi riêng duy nhất, nên chúng thường không tương thích lẫn nhau. Ngoài ra cũng có những bộ vi điều khiển 16 bits và 32 bits được sản xuất bởi các hãng khác nhau.

PIC là một họ vi điều khiển RISC được sản xuất bởi công ty Microchip Technology. Dòng PIC đầu tiên là PIC1650 được phát triển bởi Microelectronics Division thuộc General Instrument.

PIC bắt nguồn là chữ viết tắt của "Programmable Intelligent Computer" (Máy tính khả trình thông minh) là một sản phẩm của hãng General Instrument đặt cho dòng sản phẩm đầu tiên của họ là PIC1650.

Năm 1985 General Instrument bán bộ phận vi điện tử của họ, và chủ sở hữu mới hủy bỏ hầu hết các dự án - lúc đó đã quá lỗi thời. Tuy nhiên PIC được bổ sung EEPROM để tạo thành 1 bộ điều khiển vào ra khả trình. Ngày nay rất nhiều dòng PIC được xuất xưởng với hàng loạt các module ngoại vi tích hợp sẵn (như USART, PWM, ADC...), với bộ nhớ chương trình từ 512 Word đến 32K Word.



Hình 1.1: Hình dạng thực tế của PIC16F887.

1.2. PHÂN LOẠI VI ĐIỀU KHIỂN

- Phân loại theo độ dài thanh ghi

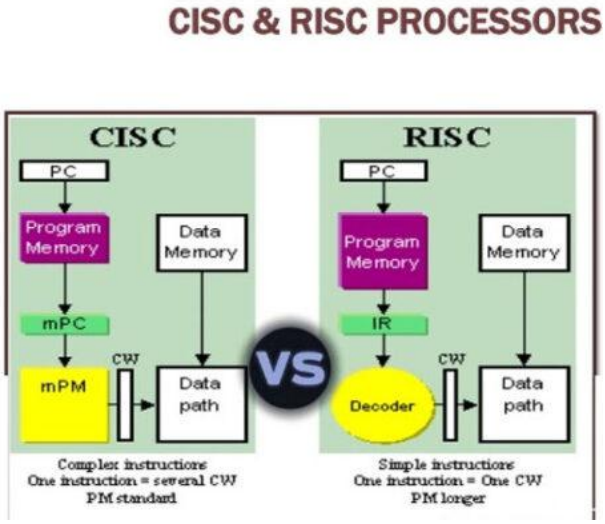
Dựa vào độ dài của các thanh ghi và các lệnh của VĐK mà người ta chia ra các loại vi điều khiển 8-bit, 16-bit hay 32-bit …

Các loại VĐK 16 bit do có độ dài lệnh lớn hơn nên các tập lệnh cũng nhiều hơn, phong phú hơn. Tuy nhiên bất cứ chương trình nào viết băng VĐK 16-bit chúng ta đều có thể viết trên vi điều khiển 8-bit với chương trình thích hợp.

- Phân loại theo kiến trúc CISC và RISC

Vi điều khiển CISC là vi điều khiển có tập lệnh phức tạp. Các VĐK này có một số lượng lớn các lệnh nên giúp cho người lập trình có thể linh hoạt và dễ dàng hơn khi viết chương trình.

Vi điều khiển RISC là vi điều khiển có tập lệnh đơn gian. Chúng có một số lượng nhỏ các lệnh đơn giản. Do đó, chúng đòi hỏi phần cứng ít hơn, giá thành thấp hơn, và nhanh hơn so với CISC. Tuy nhiên nó đòi hỏi người lập trình phải viết các chương trình phức tạp hơn, nhiều lệnh hơn.

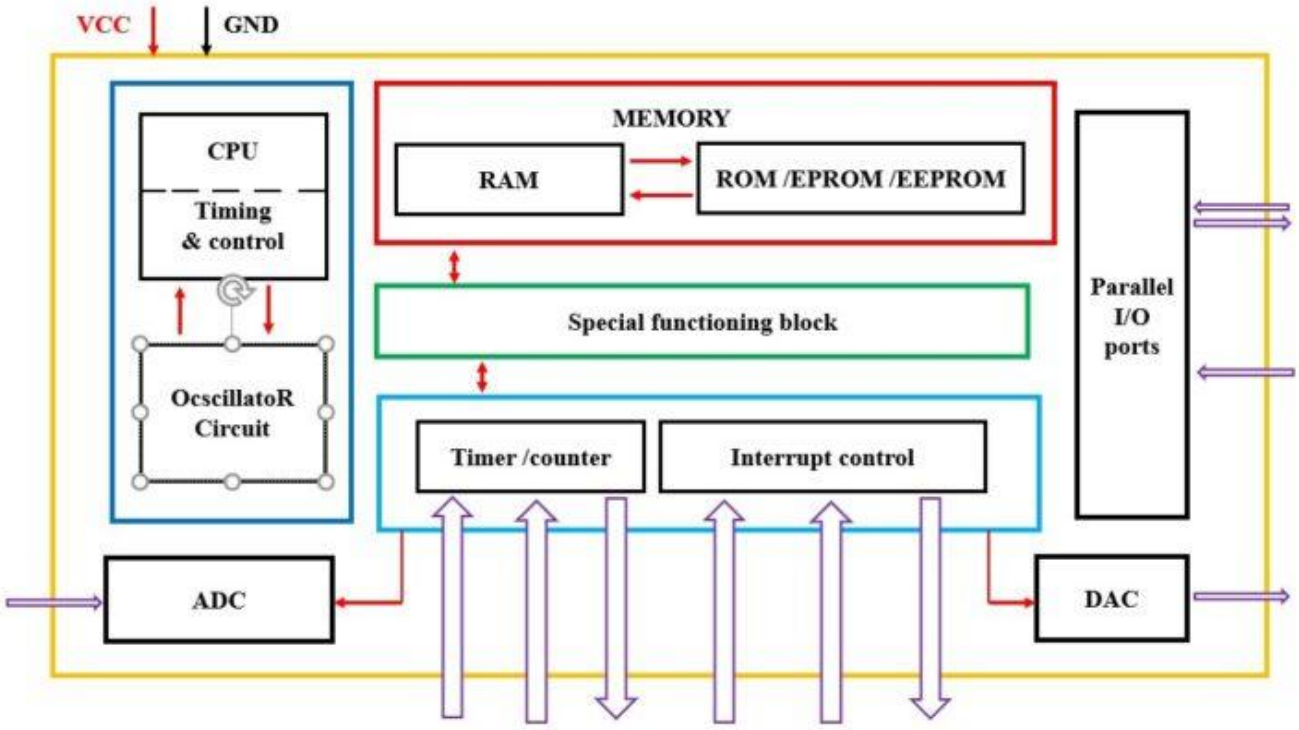


Hình 1.2: Hai kiến trúc xử lí chính

Kiến trúc Harvard và kiến trúc Von-Neumann

Kiến trúc Harvard sử dụng bộ nhớ riêng biệt cho chương trình và dữ liệu. Bus địa chỉ và bus dữ liệu độc lập với nhau nên quá trình truyền nhận dữ liệu đơn giản hơn. Kiến trúc Von-Neumann sử dụng chung bộ nhớ cho chương trình và dữ liệu. Điều này làm cho VĐK gọn nhẹ hơn, giá thành rẻ hơn.

1.3. Cấu trúc vi điều khiển



Hình 1.3: Cấu trúc tổng quan của vi điều khiển.

CPU hay Vi xử lý

CPU (Center Programing Unit) hay bộ xử lý trung tâm là bộ não của vi điều khiển. CPU chịu trách nhiệm nạp lệnh, giải mã và thực thi. Tất cả những hành vi của vi điều khiển đều là do CPU điều khiển.

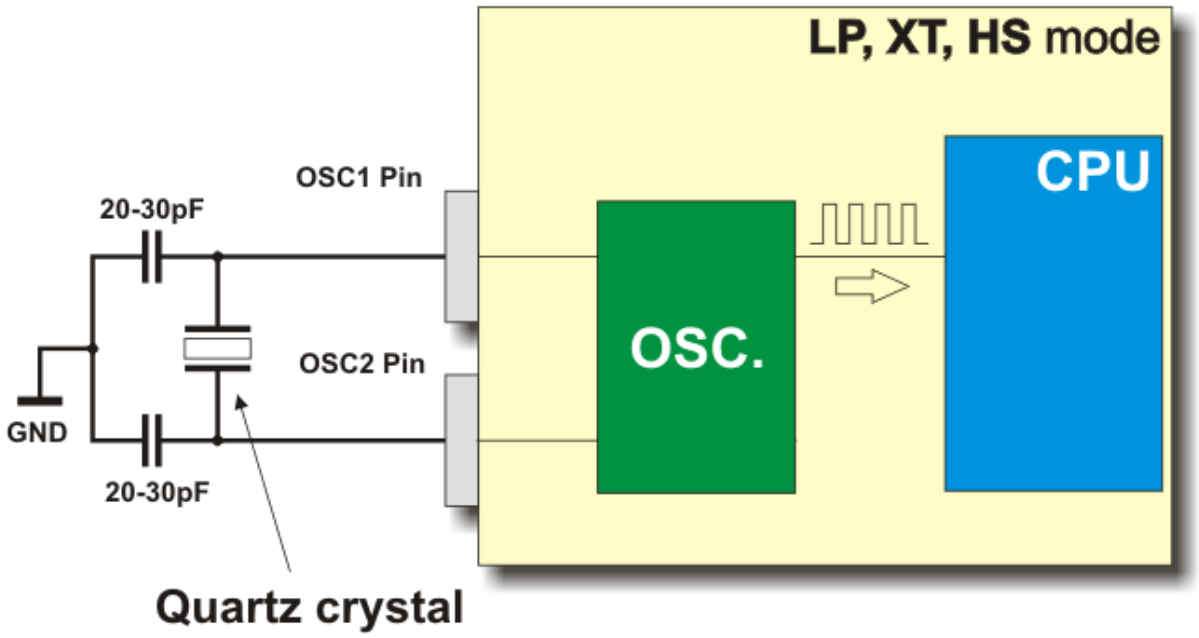
Chúng giao tiếp với các phần khác trong vi điều khiển thông qua hệ thống Bus.

Hình 1.4: CPU thường thấy

Ocscillator Circuit

Nếu CPU là bộ não thì Ocscillator Circuit hay còn gọi là Clock được coi là trái tim của vi điều khiển. Để mọi thứ có thể hoạt động, bắt buộc chúng ta phải cấp xung, trái tim hoạt động mới có thể bơm máu cho toàn bộ cơ thể hoạt động được.

Chúng ta thường nghe quảng cáo dòng vi xư lý có tốc độ bao nhiêu Ghz gì gì đó, chính là tốc độ Clock mà vi xử lý đó có thể đáp ứng được, tốc độ xung càng cao thì tốc độ xử lý của CPU cũng tăng lên. Đương nhiên mọi thứ đều có giới hạn của nó.

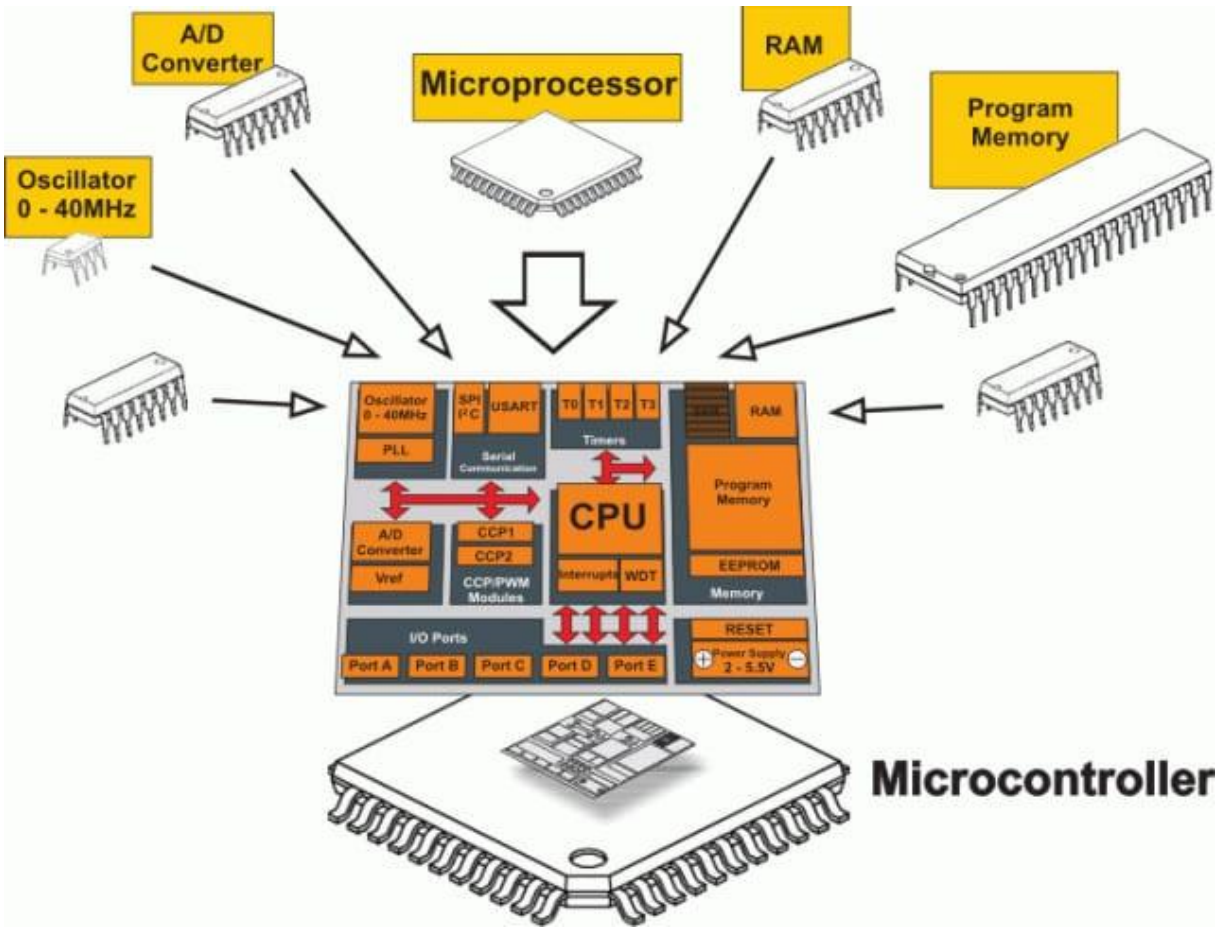


Hình 1.5: Bộ tạo xung thạch anh.

Memory – Bộ nhớ

Bộ nhớ có thể coi là một phần không thể thiếu, chúng là nơi lưu trữ chương trình nạp lên hoặc dùng làm nơi chứa các thông tin tức thời mà CPU cần dùng tới. Có 2 kiểu bộ nhớ cơ bản:

- RAM (Random access memory) là bộ nhớ lưu các dữ liệu mà CPU cần dùng để tính toán, đưa ra quyết định, chúng sẽ bị xóa khi mất điện
- ROM/EPROM/EEPROM hoặc Flash: là bộ nhớ lưu trữ chương trình hay trí khôn của vi điều khiển, chúng được ghi khi chúng ta nạp chương trình vào vi điều khiển, không bị mất khi tắt điện hoặc reset.

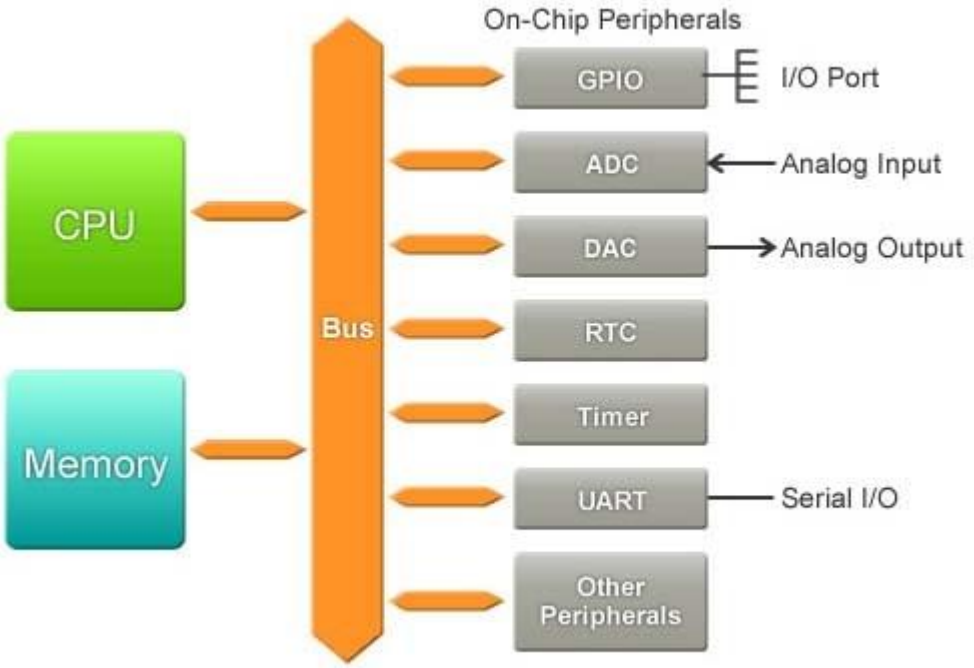


Hình 1.6: Bộ nhớ RAM/ROM tích hợp bên trong VĐK

Timer/counter

Một vi điều khiển có thể có nhiều bộ đếm thời gian và bộ đếm. Bộ đếm thời gian và bộ đếm có chức năng đếm thời gian tạo ra các sự kiện để vi điều khiển hoạt động đúng thời điểm.

Các ngoại vi của vi điều khiển



Hình 1.7: Các chức năng ngoại vi của VDK.

I/O Ports – Input/ouput

Có thể coi I/O Port là tay chân của vi điều khiển, chúng giúp cho vi điều khiển tương tác với các thành phần khác ngoài môi trường.

Cổng đầu vào / đầu ra được sử dụng chủ yếu điều khiển hoặc giao tiếp các thiết bị như màn hình LCD, đèn LED, máy in, …cho vi điều khiển.

Các chuẩn giao tiếp

Giống như miệng và tai vậy. Vi điều khiển sẽ sử dụng các chuẩn giao tiếp khác nhau để liên lạc với nhau hoặc liên lạc với các phâng tử khác trên mạch. Có thể kể đến như I2C, SPI, UART, USB, ….

Bộ chuyển đổi analog sang digital (ADC)

Bộ chuyển đổi ADC được sử dụng để chuyển đổi tín hiệu analog sang dạng digital. Tín hiệu đầu vào trong bộ chuyển đổi này phải ở dạng analog (ví dụ: đầu ra cảm biến) và đầu ra từ thiết bị này ở dạng digital. Đầu ra digital có thể được sử dụng cho các ứng dụng kỹ thuật số (ví dụ: các thiết bị đo lường).

Bộ chuyển đổi Digital sang Analog (DAC)

Hoạt động của DAC là đảo ngược của ADC. DAC chuyển đổi tín hiệu digital thành định dạng analog. Nó thường được sử dụng để điều khiển các thiết bị analog như động cơ DC, các ổ đĩa…

Interrupt control hay quản lý sự kiện

Ngoài việc thực thi chương trình, vi điều khiển còn phải tương tác với các tác nhân bên trong và bên ngoài. Các tác nhân này sẽ tạo ra các sự kiện gọi là Ngắt, để quản lý nó cần có một khối quản lý ngắt (Interrupt control)

1.4. Nguyên lý hoạt động

Nguyên lý hoạt động của vi điều khiển có thể được chia thành các bước chính sau:

- Chu trình lấy lệnh, giải mã và thực thi

+ Lấy lệnh (Fetch): Vi điều khiển đọc lệnh từ bộ nhớ chương trình (ROM/Flash).

+ Giải mã (Decode): Bộ điều khiển giải mã lệnh để xác định cần thực hiện thao tác gì.

+ Thực thi (Execute): Vi điều khiển thực hiện lệnh bằng cách truy cập bộ nhớ, điều khiển các thiết bị ngoại vi hoặc tính toán.

- Xử lý tín hiệu vào/ra (I/O)

+ Vi điều khiển nhận tín hiệu từ các cảm biến, công tắc, bàn phím, v.v.

+ Sau đó, nó xử lý tín hiệu và gửi lệnh điều khiển ra các thiết bị như LED, động cơ, màn hình hiển thị.

- Hoạt động của bộ nhớ

+ Bộ nhớ chương trình (ROM/Flash): Lưu trữ chương trình điều khiển.

+ Bộ nhớ dữ liệu (RAM): Lưu trữ tạm thời dữ liệu trong quá trình xử lý.

+ Bộ nhớ EEPROM (nếu có): Lưu trữ dữ liệu vĩnh viễn, ngay cả khi mất nguồn.

- Quản lý ngắt (Interrupt)

+ Khi có sự kiện xảy ra (như tín hiệu từ cảm biến), vi điều khiển tạm dừng chương trình chính để xử lý sự kiện, sau đó quay lại tiếp tục chương trình.

- Giao tiếp ngoại vi

+ Vi điều khiển có thể giao tiếp với các thiết bị khác thông qua các giao thức như UART, SPI, I2C để truyền và nhận dữ liệu.

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG

2.1. Mô tả hệ thống

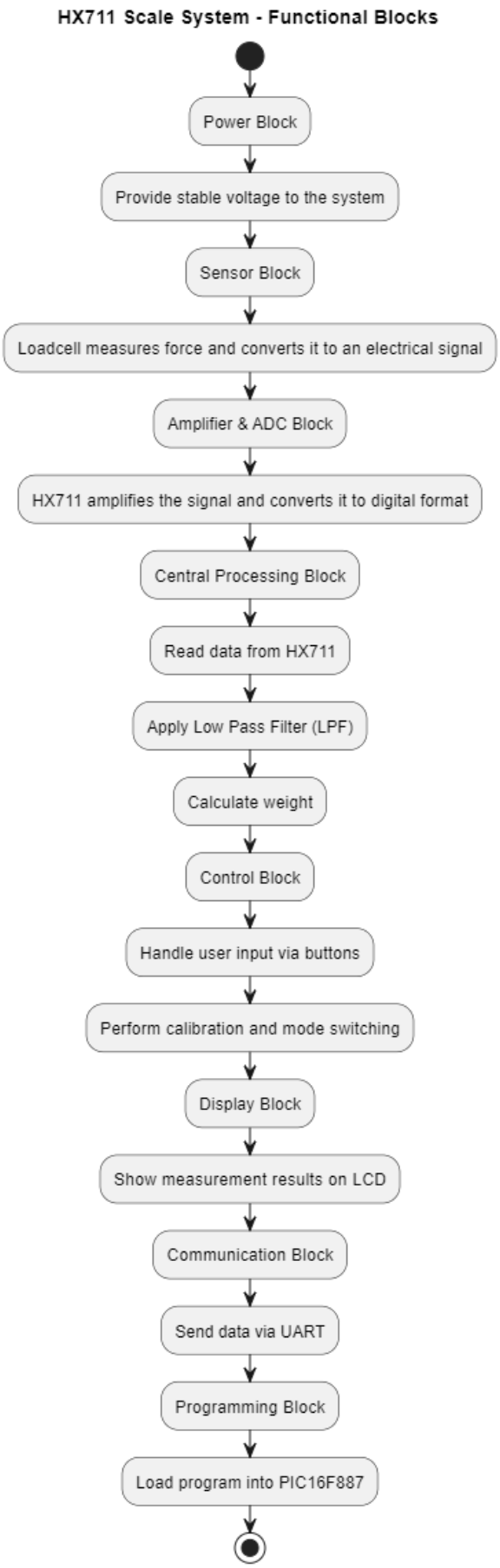
Trong chương này, nhóm sẽ trình bày thiết kế **hệ thống cân điện tử** sử dụng **cảm biến loadcell** kết hợp với **mạch khuếch đại HX711** và **thuật toán Low Pass Filter (LPF)**. Hệ thống bao gồm cả phần cứng và phần mềm để đo khối lượng một cách chính xác, giảm nhiễu tín hiệu bằng thuật toán LPF, giúp cải thiện độ ổn định của kết quả đo.

2.2 Thiết kế sơ đồ khối

Hệ thống cân điện tử được thiết kế dựa trên các khối chức năng chính sau:

- Khối nguồn: Cung cấp điện áp ổn định cho toàn bộ hệ thống.
- Khối cảm biến: Loadcell đo lực tác động và chuyển đổi thành tín hiệu điện.
- Khối khuếch đại & chuyển đổi ADC: Mạch HX711 khuếch đại tín hiệu từ loadcell và chuyển đổi sang dạng số để vi điều khiển xử lý.
- Khối xử lý trung tâm: Vi điều khiển PIC16F887 đọc dữ liệu từ HX711, xử lý tín hiệu và áp dụng thuật toán LPF để giảm nhiễu.
- Khối giao tiếp: Kết nối và hiển thị dữ liệu lên màn hình LCD.
- Khối lập trình (Programmer): Nạp chương trình vào vi điều khiển PIC16F887.
- Khối nút nhấn: Cung cấp các chức năng điều khiển cho người dùng như hiệu chỉnh cân.
- Khối hiển thị: Hiển thị kết quả đo lường trên màn hình LCD.
- Khối điều khiển: Thực hiện các chế độ đo và hiệu chỉnh cân thông qua nút bấm.

Hệ thống này giúp đảm bảo việc đo khối lượng chính xác, giảm sai số nhờ bộ lọc LPF, đồng thời cung cấp giao diện đơn giản để hiển thị kết quả đo dễ dàng.



Hình 2.1: *Sơ đồ khối hệ thống.*

2.3.1. Sơ đồ khối mạch nguồn

Khối nguồn đóng vai trò rất quan trọng trong hệ thống, cung cấp điện áp ổn định để đảm bảo hoạt động của khối xử lý trung tâm, khối cảm biến, khối nút bấm và khối hiển thị.

Hệ thống hoạt động chủ yếu ở mức điện áp +5V DC, do đó nguồn nuôi cần phải được thiết kế để đảm bảo hiệu suất cao, ổn định và tránh sụt áp trong quá trình vận hành. Hiện nay, có hai phương pháp chính để cung cấp nguồn:

- Dùng pin hoặc ắc quy:

  + Ưu điểm: Điện áp đầu ra tương đối ổn định.

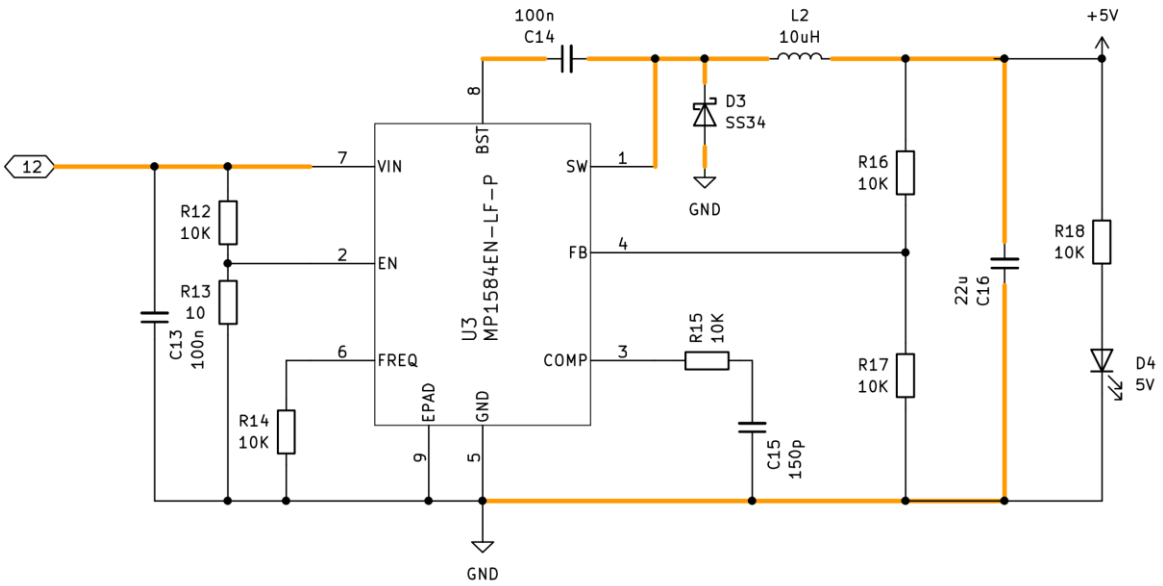
  + Nhược điểm: Không có loại pin hoặc ắc quy nào cung cấp chính xác 5V, do đó cần có mạch biến đổi điện áp để điều chỉnh về mức điện áp chuẩn. Ngoài ra, khi dung lượng pin giảm, điện áp có thể bị tụt, gây gián đoạn hoạt động của hệ thống.

- Dùng mạch chuyển đổi điện áp từ nguồn cao hơn (7-9V DC) xuống 5V DC:

  + Trong thiết kế này, hệ thống sử dụng nguồn 7-9V DC, sau đó qua mạch hạ áp (buck converter) để giảm xuống 5V DC.

  + So với mạch hạ áp tuyến tính (linear regulator), mạch hạ áp buck có hiệu suất cao hơn (có thể đạt 90%), hỗ trợ dòng tải lên đến 3A, đồng thời giảm tổn hao nhiệt – một nhược điểm lớn của mạch tuyến tính.

Sơ đồ mạch chi tiết được minh họa trong hình 2.2.



Hình 1.2: *Sơ đồ nguyên lý mạch nguồn*

Mô tả chân của mạch hạ áp (Buck Converter)

1. Nguồn vào (VIN)

- Chân 7 (VIN): Nhận điện áp đầu vào từ 7-9V DC, cung cấp nguồn cho toàn bộ mạch điều khiển bên trong IC.

2. Chân kích hoạt (EN - Enable Input)

Chân 2 (EN): Điều khiển bật/tắt IC.

- Khi chân này được kéo lên mức cao thông qua điện trở R5 (100kΩ) và R6 (30kΩ), IC sẽ hoạt động.

- Nếu kéo chân này xuống dưới mức ngưỡng quy định, IC sẽ tắt.

3. Chân chuyển mạch (SW - Switch Node)

- Chân 1 (SW): Đầu ra từ bộ chuyển mạch high-side, kết nối với cuộn cảm L1 (15µH) và diode D2 (SS34).

- Diode này được đặt gần chân SW để giảm thiểu nhiễu do quá trình chuyển mạch.

4. Chân phản hồi (FB - Feedback)

- Chân 4 (FB): Nhận tín hiệu phản hồi để điều chỉnh điện áp đầu ra.

- Điện áp phản hồi được thiết lập thông qua mạng phân áp điện trở gồm R8 (51kΩ) và R9 (10kΩ).

- Điện áp chuẩn nội bộ của IC là 0.8V.

5. Chân bù (COMP - Compensation)

- Chân 3 (COMP): Đầu ra của bộ khuếch đại lỗi, dùng để ổn định hệ thống phản hồi.

- Điện áp tại chân này được điều chỉnh bởi R10 (100kΩ) và tụ C11 (150pF) để đảm bảo hiệu suất ổn định.

6. Chân thiết lập tần số (FREQ - Switching Frequency Program Input)

- Chân 6 (FREQ): Xác định tần số chuyển mạch của IC.

- Điện trở R7 (100kΩ) được kết nối từ chân này xuống đất để thiết lập tần số hoạt động của IC.

7. Chân nối đất (GND - Ground)

- Chân 5 (GND): Kết nối với mát (ground) của mạch.

- Để giảm trở kháng và nhiễu, chân này nên được nối gần với tụ điện đầu ra.

8. Chân Bootstrap (BST)

- Chân 8 (BST): Cung cấp nguồn dương cho driver MOSFET high-side bên trong IC.

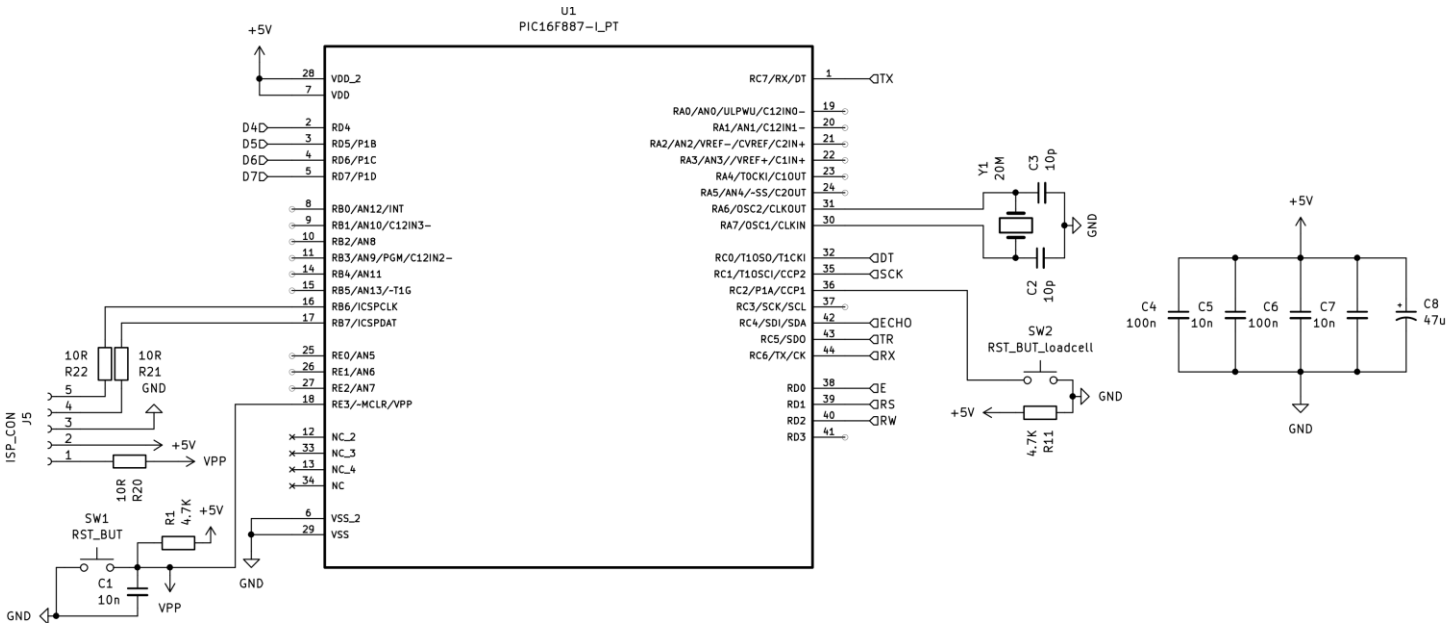
- Một tụ điện bootstrap C10 (100nF) được kết nối giữa chân BST và chân SW để hỗ trợ quá trình đóng/mở MOSFET.

9. Tụ lọc đầu vào và đầu ra

- Tụ C9 (100nF): Giúp ổn định điện áp đầu vào, giảm nhiễu.

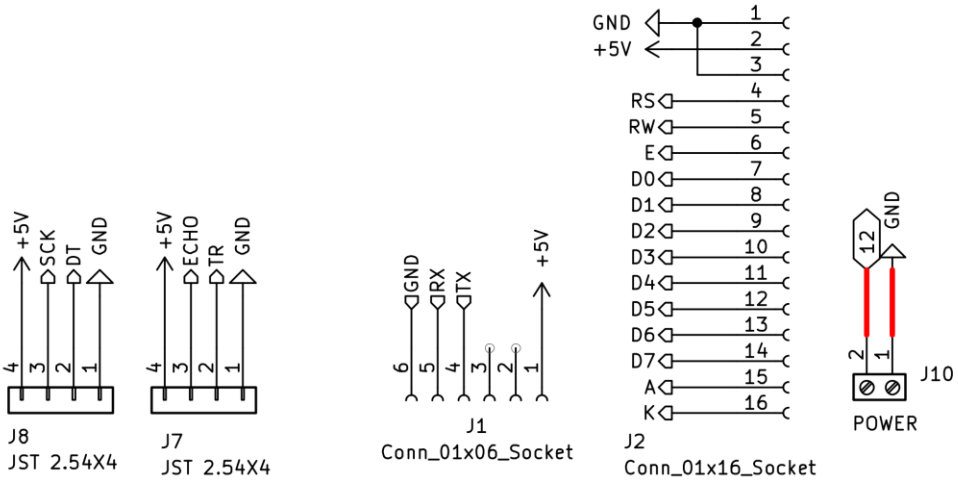
- Tụ C12 (22µF): Được sử dụng ở đầu ra để duy trì điện áp ổn định và giảm nhiễu tần số cao.

2.3.2. Khối MCU



Hình 2.3: *Sơ đồ khối MCU*.

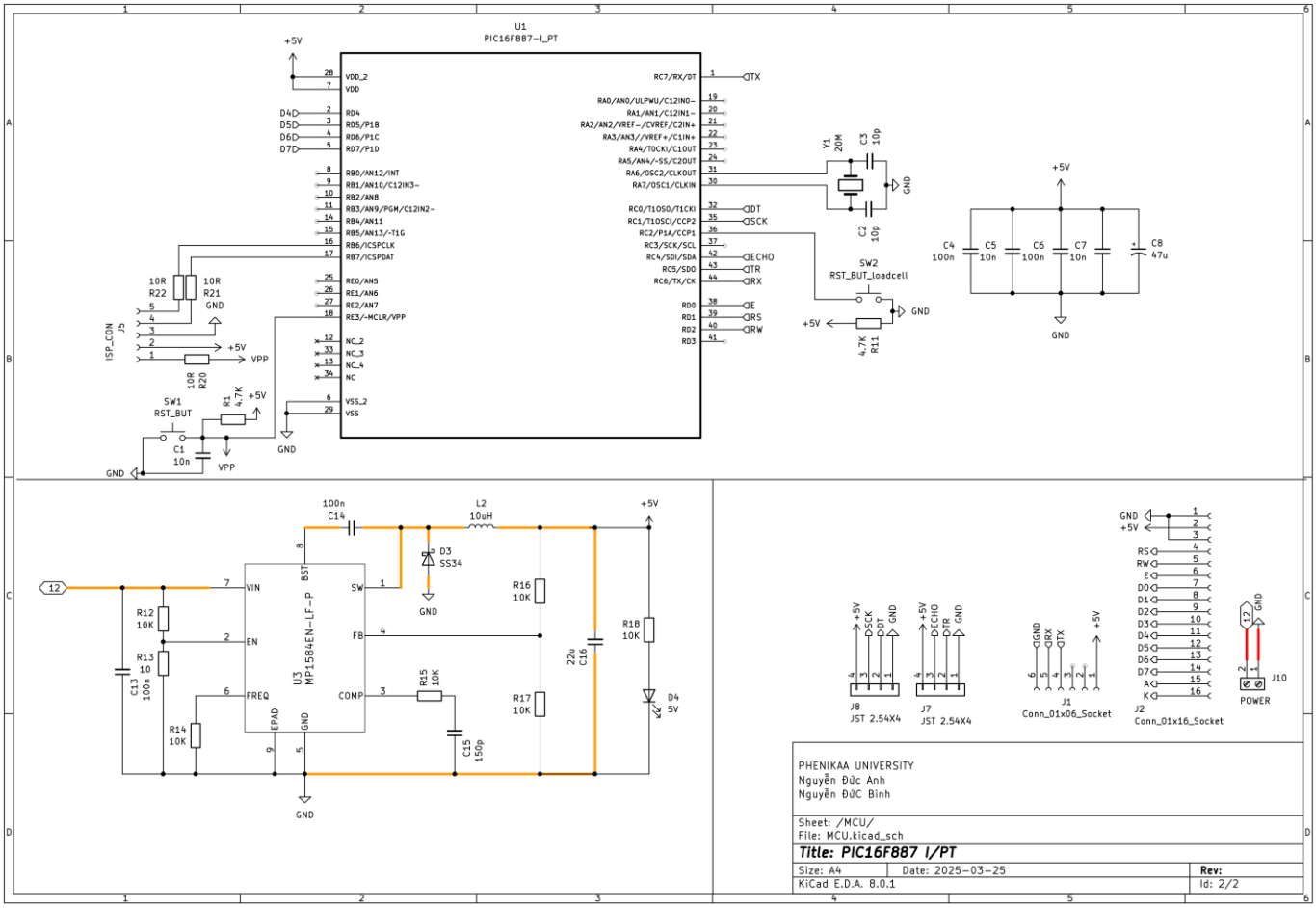
2.3.3. Sơ đồ khối Header



Hình 2.4: Sơ đồ khối header.

2.3. Thiết kế sơ đồ nguyên lý

2.3.1. Sơ đồ nguyên lý tổng quan



Hình 2.5: *Sơ đồ nguyên lý*

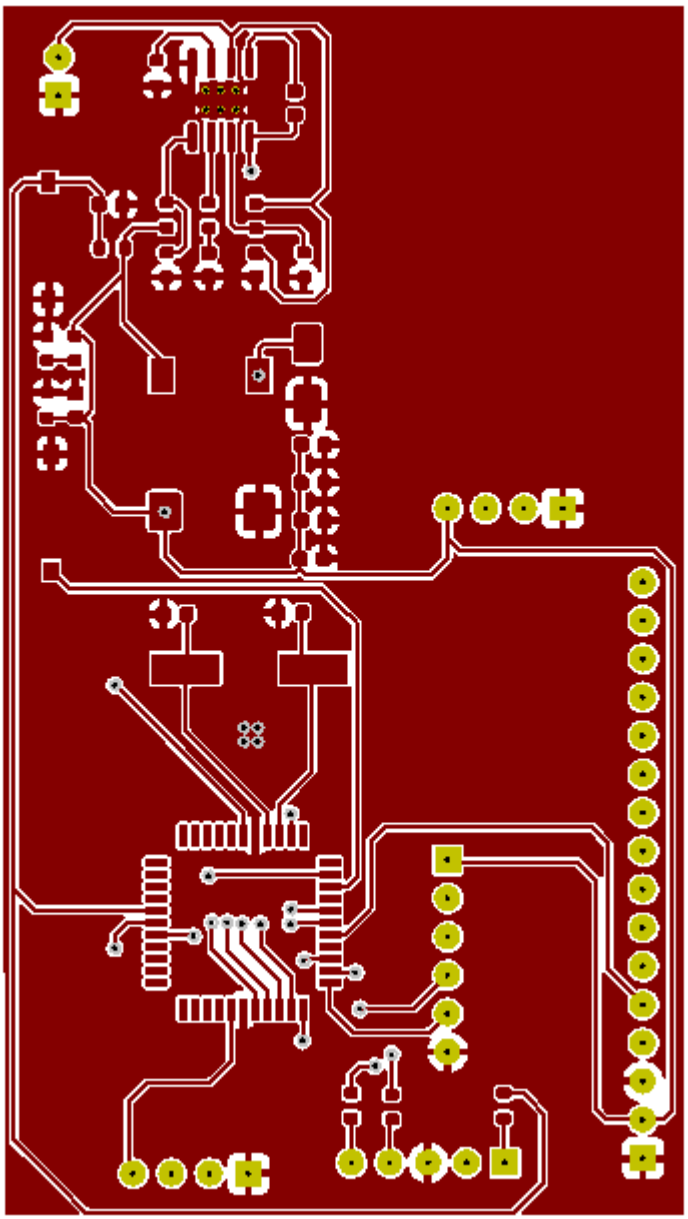
Bảng 2.6: Bảng linh kiện và tham số linh kiện sử dụng trong mạch

Reference	**Value**	**Datasheet**	Footprint	Qty
C1,C5,C7	10n	~	Capacitor_SMD: C_0603_1608Metric	3
C2,C3	10p	~	Capacitor_SMD: C_0603_1608Metric	2
C4,C6,C13,C14	100n	~	Capacitor_SMD: C_0603_1608Metric	4
C8	47u	~	Capacitor_Tantalum _SMD:CP_ EIA-7343-30_AVX-N	1
C15	150p	~	Capacitor_SMD: C_0603_1608Metric	1
C16	22u	~	Capacitor_SMD: C_0603_1608Metric	1

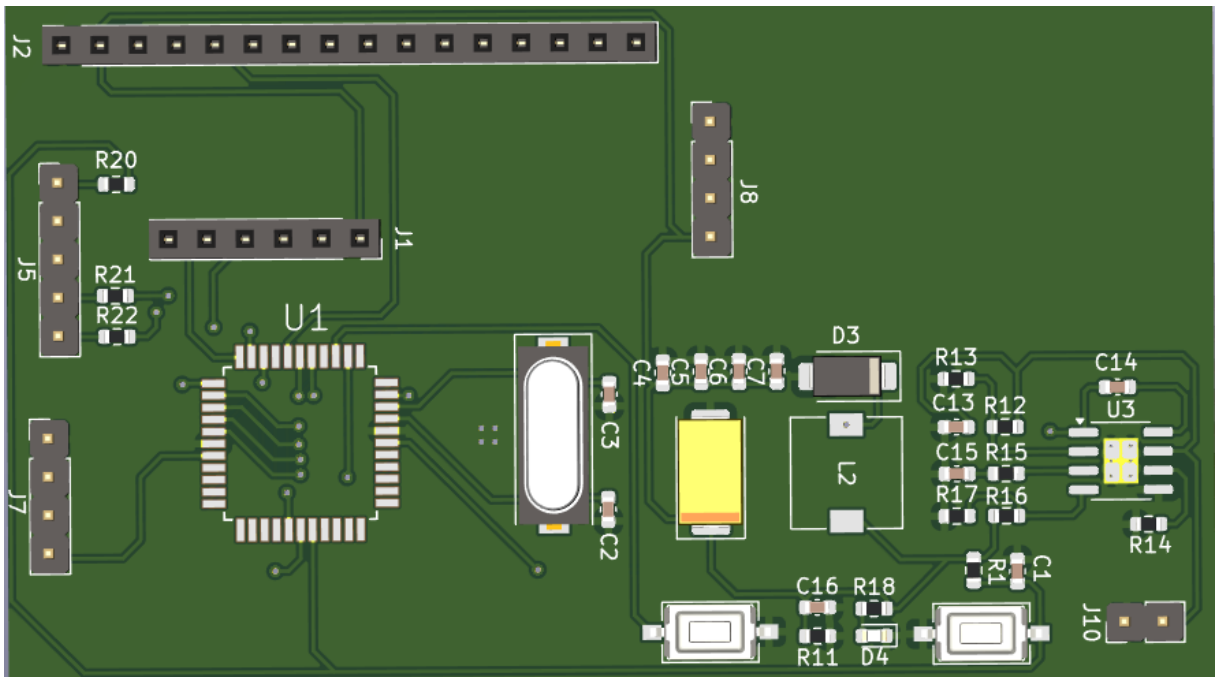
D3	SS34	https://www.microsemi.com/documentportal/doc_download/131890-lds-0040-1-datasheet	Diode_SMD:D_SMA	1
D4	5V	~	LED_SMD:LED_0603_1608Metric	1
J1	Conn_01x06_Socket	~	Connector_PinSocket_2.54mm:PinSocket_1x06_P2.54mm_Vertical	1
J2	Conn_01x16_Socket	~	Connector_PinSocket_2.54mm:PinSocket_1x16_P2.54mm_Vertical	1
J5	ISP_CON	~	Connector_PinHeader_2.54mm:PinHeader_1x05_P2.54mm_Vertical	1
J7,J8	JST 2.54X4	~	Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical	2
J10	POWER	~	Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical	1
L2	10uH	~	Inductor_SMD:L_7.3x7.3_H3.5	1
R1,R11	4.7K	~	Resistor_SMD:R_0603_1608Metric	2
R12,R14,R15 R16,R17,R18	10K	~	Resistor_SMD:R_0603_1608Metric	7
R20,R21,R22	10R	~	Resistor_SMD:R_0603_1608Metric	3
SW1	RST_BUT	~	Button_Switch_SMD:SW_Tactile_SPST_NO_Straight_CK_PTS636Sx25SMTRLFS	1
SW2	RST_BUT_loadcell	~	Button_Switch_SMD:SW_Tactile_SPST_NO_Straight_CK_PTS636Sx25SMTRLFS	1
U1	PIC16F887-I_PT		Package_QFP:QFP80P1200X1200X120-44N	1
U3	MP1584EN-LF-P	MP1584EN-LF-P	Package_SO:SOIC-8-1EP_3.9x4.9mm_P1.27mm_EP2.29x3mm_ThermalVias	1
Y1	20M	~	Crystal:	1

			Crystal_SMD_HC49-SD	

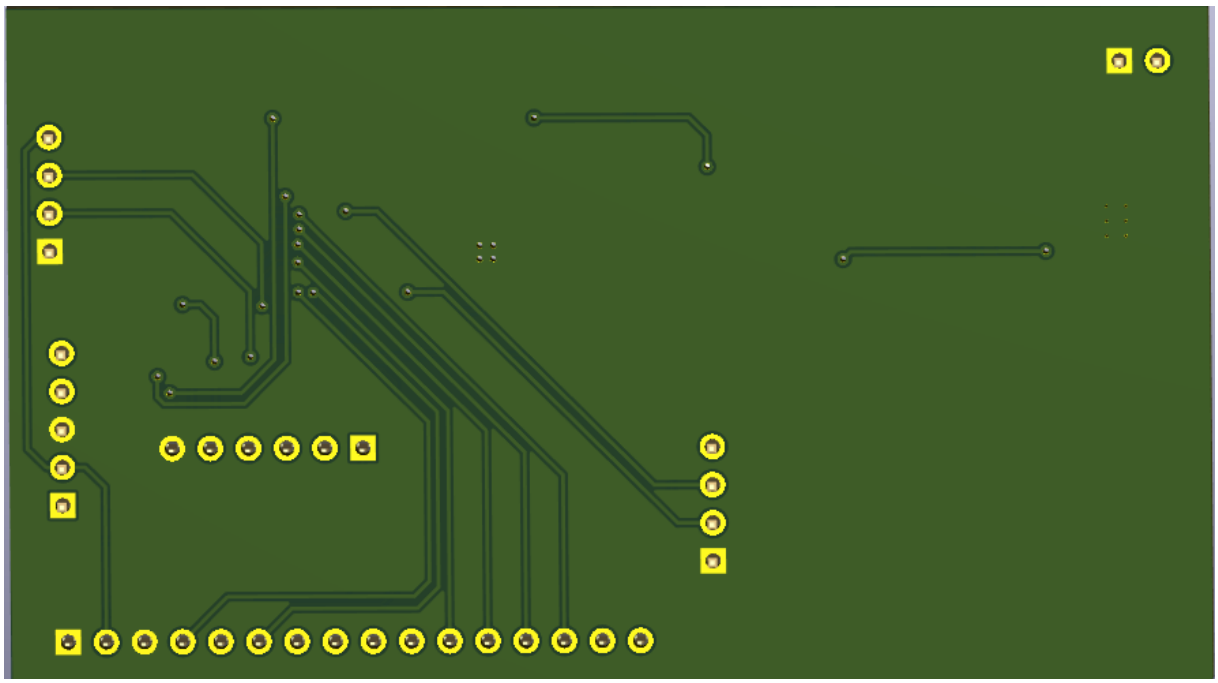
2.4. Thiết kế layout và chế tạo PCB



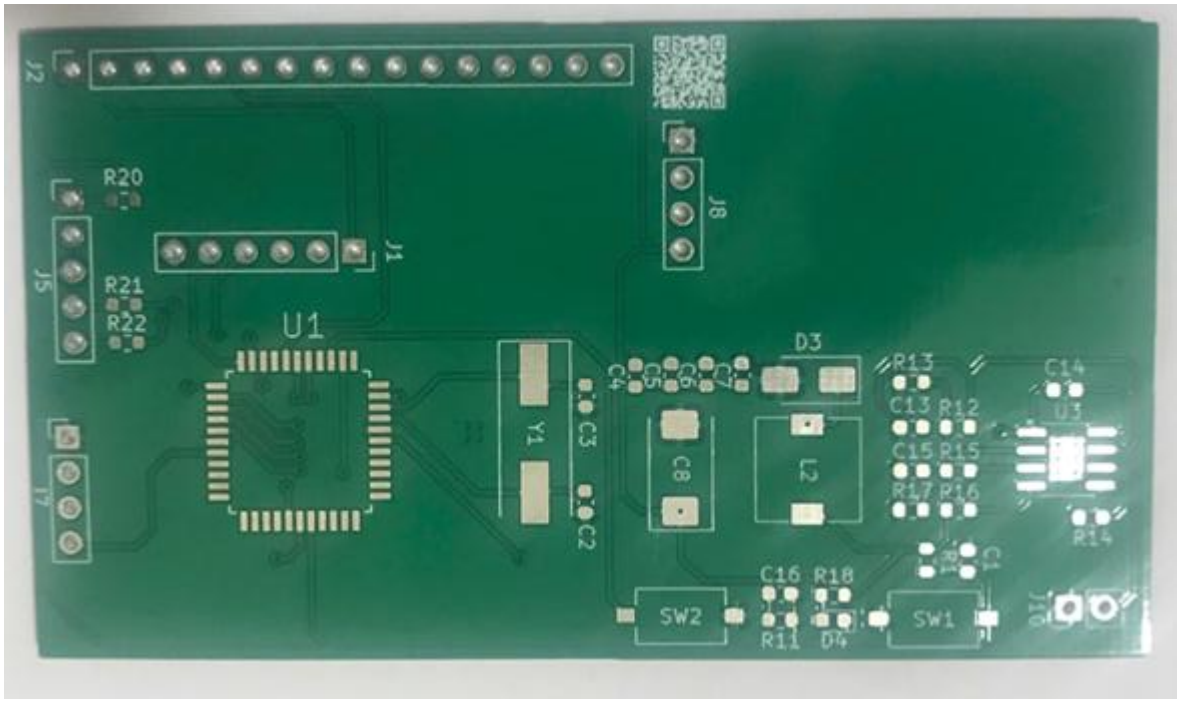
Hình 2.6.a: *Layout Mạch dán PCB được thiết kế bằng ứng dụng Kicad*.



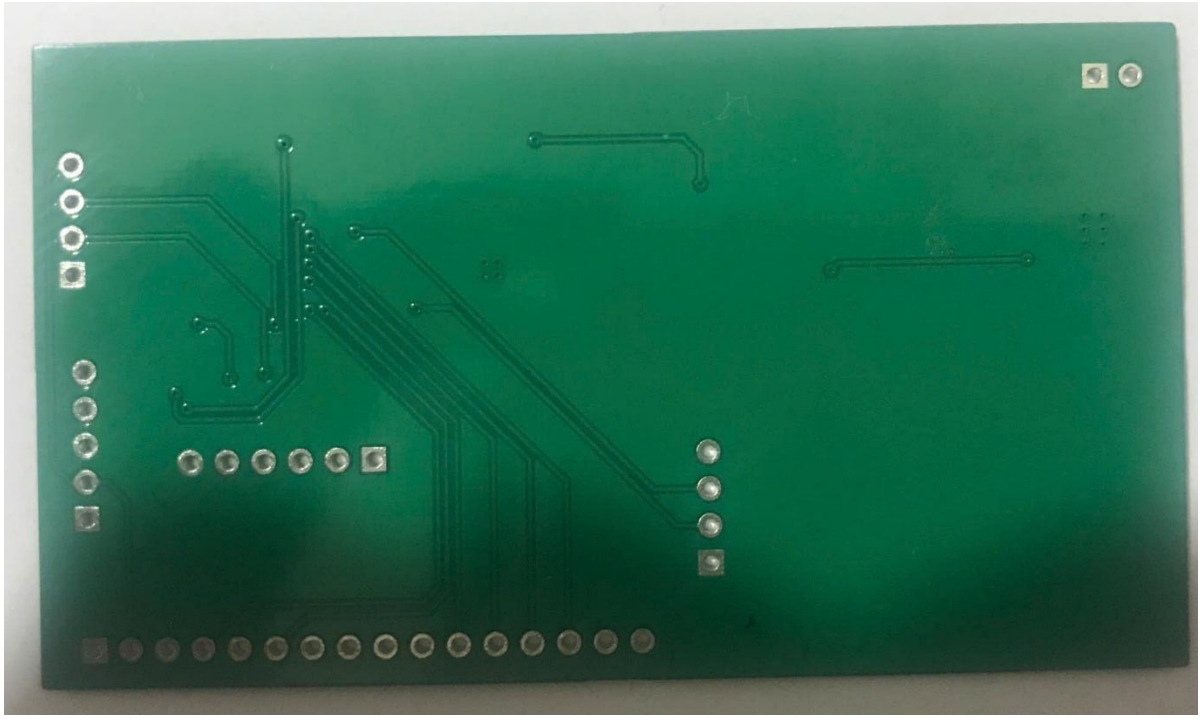
Hình 2.6.b: *Mạch dán PCB mặt trước được thiết kế bằng ứng dụng Kicad.*



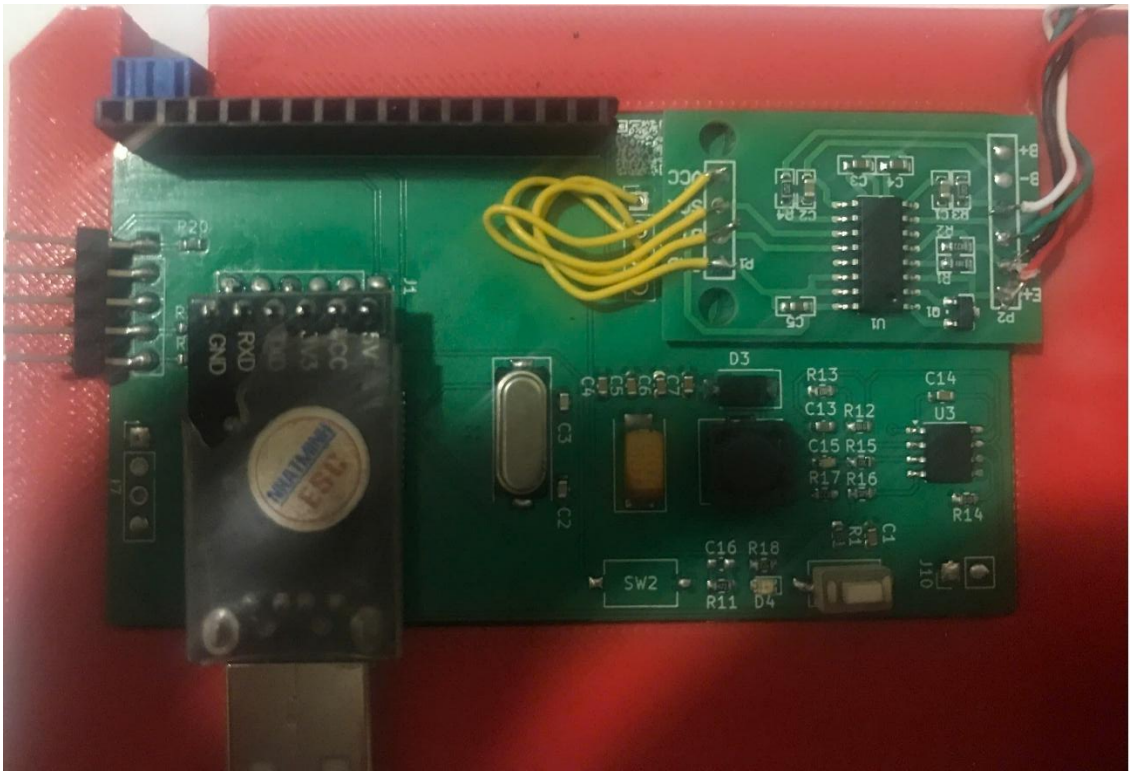
Hình 2.6.c: *Mạch dán PCB mặt sau được thiết kế bằng ứng dụng Kicad.*



Hình 2.6.d: *Mạch dán PCB mặt trước thực tế*



Hình 2.6.e: *Mạch dán PCB mặt sau thực tế*



Hình 2.6.f: *Mạch đã có linh kiện*

2.5. Thiết kế giải thuật

2.5.1. Thuật toán đọc bit data

Trong sản phẩm của nhóm em có modul HX711 là một bộ chuyển đổi tín hiệu ADC 24-bit chuyên dụng dùng để đọc dữ liệu từ các cảm biến lực (load cell). Nó hoạt động bằng cách chuyển đổi tín hiệu điện áp rất nhỏ từ load cell thành dữ liệu số có độ chính xác cao. Cần phải đọc 24 bit dữ liệu từ bộ chuyển đổi sang vi xử lí cần có thuật toán để đọc.

```
unsigned int32 readCount(void)
{
   unsigned int32 data = 0;
   unsigned int8 j;

   output_bit(DT1, 1);
   output_bit(SCK, 0);
   while (input(DT1))
      ;

   for (j = 0; j < 24; j++)
   {
      output_bit(SCK, 1);
      data = data << 1;
```

```
            output_bit(SCK, 0);
            if (input(DT1))
            {
               data++;
            }
         }

         output_bit(SCK, 1);
         data = data ^ 0x800000;
         output_bit(SCK, 0);
         return data;
      }
```

Cách thức hoạt động của thuật toán:

- Chuẩn bị dữ liệu

+ Đặt DT1 = 1 (không bắt buộc, thường để đảm bảo trạng thái cao ban đầu).
+ Đặt SCK = 0 để sẵn sàng đọc dữ liệu.
+ Chờ đến khi DT1 chuyển xuống mức thấp, tức là HX711 đã sẵn sàng gửi dữ liệu.

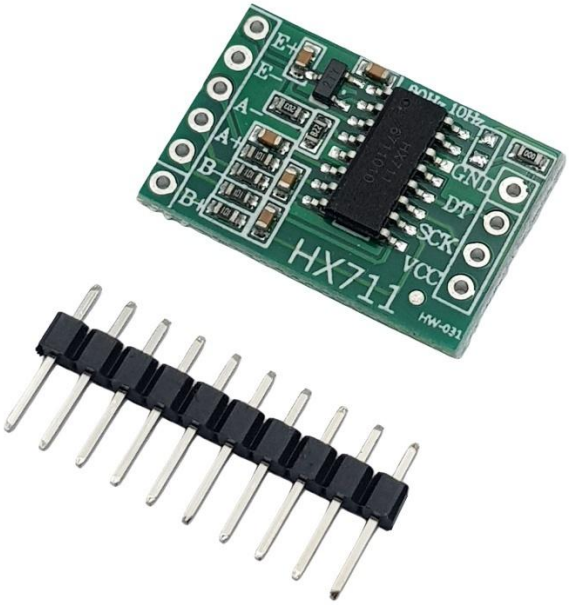
- Đọc dữ liệu 24-bit từ HX711

+ Lặp 24 lần để lấy từng bit của dữ liệu.
+ Dịch trái (data = data << 1) để tạo chỗ cho bit mới.
+ Kéo SCK lên (output_bit(SCK, 1)) để lấy bit tiếp theo.
+ Nếu DT1 đang ở mức cao, bit mới được thêm vào dữ liệu (data++).
+ Kéo SCK xuống (output_bit(SCK, 0)) để hoàn thành chu kỳ đọc bit.

- Định dạng lại dữ liệu

+ Sau khi đọc xong 24-bit, gửi một xung nhịp (SCK = 1) để HX711 chuẩn bị cho lần đọc tiếp theo.
+ Thực hiện phép XOR với 0x800000 để đảm bảo dữ liệu được đúng dạng có dấu (two’s complement).

Trả về dữ liệu đã đọc.



Hình 2.7.a : *Mạch chuyển đổi ADC 24-bit loadcell*

Hình 2.7.b : *Loadcell*

2.5.2. Thuật toán lọc nhiễu LPF

Thuật toán Low-Pass Filter (LPF) có nhiều loại tuy nhiên trong dự án này em sử dụng với bộ lọc trung bình cộng (SMA).

Bộ lọc thông thấp (Low-Pass Filter - LPF) là một thuật toán được sử dụng để loại bỏ nhiễu cao tần và giữ lại các thành phần tần số thấp của tín hiệu. LPF thường được áp dụng trong các hệ thống xử lý tín hiệu số, điện tử, và điều khiển nhúng để làm mượt dữ liệu, giảm sai số và cải thiện độ ổn định.

Bộ lọc Trung bình Cộng (SMA) là một dạng đơn giản của LPF, hoạt động bằng cách tính trung bình cộng của một số lượng giá trị dữ liệu gần nhất. SMA giúp làm mượt tín hiệu bằng cách giảm ảnh hưởng của nhiễu và biến động ngẫu nhiên.

Công thức tính SMA:

$$y_n = \frac{x_n + x_{n-1} + x_{n-2} + \ldots + x_{n-M} + 1}{M}$$

Trong đó:

y_n là giá trị đầu ra đã được lọc tại thời điểm m.

x_n là giá trị đầu vào tại thời điểm n.

M là kích thước cửa sổ (số mẫu được lấy trung bình).

Cách hoạt động của SMA

- Lưu trữ M giá trị đầu vào gần nhất trong một bộ đệm (buffer).
- Khi có giá trị mới, loại bỏ giá trị cũ nhất, thêm giá trị mới nhất vào buffer.
- Tính trung bình của tất cả các giá trị trong buffer để tạo đầu ra mới.
- Lặp lại quy trình trên liên tục.

Cài đặt thuật toán SMA trong C

Dưới đây là một đoạn mã C để cài đặt thuật toán SMA với cửa sổ lọc M=5 (Theo như code của chúng em)

```
#define WINDOW_SIZE 5  // Kích thước cửa sổ trung bình
float lowpass_sma(float new_sample) {
   static float buffer[WINDOW_SIZE] = {0};  // Bộ đệm lưu giá trị
   static int index = 0;  // Vị trí hiện tại trong bộ đệm
   static float sum = 0;  // Tổng các giá trị trong bộ đệm

   sum -= buffer[index];  // Loại bỏ giá trị cũ nhất khỏi tổng
   buffer[index] = new_sample;  // Cập nhật giá trị mới vào bộ đệm
   sum += new_sample;  // Thêm giá trị mới vào tổng
   index = (index + 1) % WINDOW_SIZE;  // Cập nhật vị trí lưu trong mảng

   return sum / WINDOW_SIZE;  // Trả về giá trị trung bình
}
```

Ưu điểm và Nhược điểm của SMA

- Ưu điểm:

Dễ cài đặt, tốn ít tài nguyên tính toán.

Làm mượt dữ liệu hiệu quả với tín hiệu ít biến động.

Giúp loại bỏ nhiễu nhanh chóng.

- Nhược điểm:

Phản hồi chậm với các thay đổi đột ngột trong tín hiệu.

Có thể gây trễ tín hiệu do cần lấy trung bình nhiều giá trị.

Ứng dụng của SMA trong thực tế

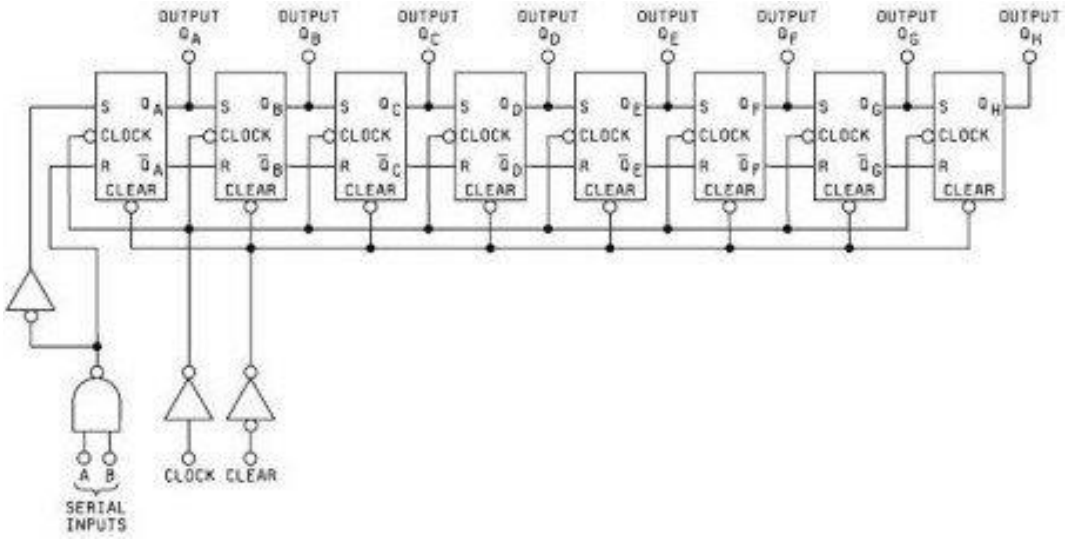
- Xử lý dữ liệu cảm biến: Giảm nhiễu từ cảm biến nhiệt độ, gia tốc kế, áp suất.
- Thị trường tài chính: Được sử dụng để làm mượt dữ liệu giá cổ phiếu.
- Điều khiển nhúng & IoT: Làm mượt tín hiệu đo từ các cảm biến điện tử.

2.5.3. Thuật toán ngoại vi UART

Các giao thức giao tiếp nối tiếp khác nhau tồn tại, mỗi giao thức có chi tiết hoạt động độc đáo. Tuy nhiên, tất cả họ đều chia sẻ một cốt lõi chung: đăng ký thay đổi. Các thanh ghi này là nền tảng trong việc triển khai phần cứng của các giao thức giao tiếp nối tiếp, chuyển dữ liệu từng bit từng chu kỳ.

Cách thức hoạt động:

Các thanh ghi dịch chuyển bao gồm các flip-flop được kết nối huyết thanh chia sẻ cùng một dòng đồng hồ. Đầu vào dữ liệu được chuyển từ chân đầu vào sang đầu đầu ra, chuyển một bit cho mỗi chu kỳ đồng hồ. Điều này có nghĩa là một thanh ghi thay đổi 8 bit mất 8 chu kỳ đồng hồ để chuyển byte.



Hình 2.8: Minh họa một thanh ghi thay đổi 8 bit với các dòng đầu vào và đầu ra nối tiếp

Giao tiếp nối tiếp là cơ bản trong nhiều ứng dụng, chẳng hạn như giao tiếp thiết bị bên ngoài, tải xuống chương trình cơ sở, I/O điều khiển, truyền dữ liệu và giao diện gỡ lỗi. Thành thạo các giao thức giao tiếp nối tiếp khác nhau là điều cần thiết cho các kỹ sư hệ thống nhúng do sử dụng rộng rãi của họ. Các giao thức phổ biến bao gồm USB, CAN, I2C, I2S, LIN, SPI, Ethernet, 1 dây và UART/USART. Hướng dẫn này tập trung vào UART để giải thích các nguyên tắc cơ bản và cơ học của nó, dẫn đến truyền dữ liệu giữa MCU nhúng.

Thế nào là UART(Universal Asynchronous Receiver/transmitter):

UART, viết tắt cho máy thu/máy phát không đồng bộ phổ quát, bao gồm mạch phần cứng cần thiết cho giao tiếp nối tiếp, có sẵn dưới dạng IC độc lập hoặc các mô - đun nội bộ trong vi điều khiển. Phương pháp giao tiếp này dựa trên các chân I/O chuyên dụng: RX (kết thúc nhận) và TX (kết thúc truyền).

Các chế độ giao tiếp UART:

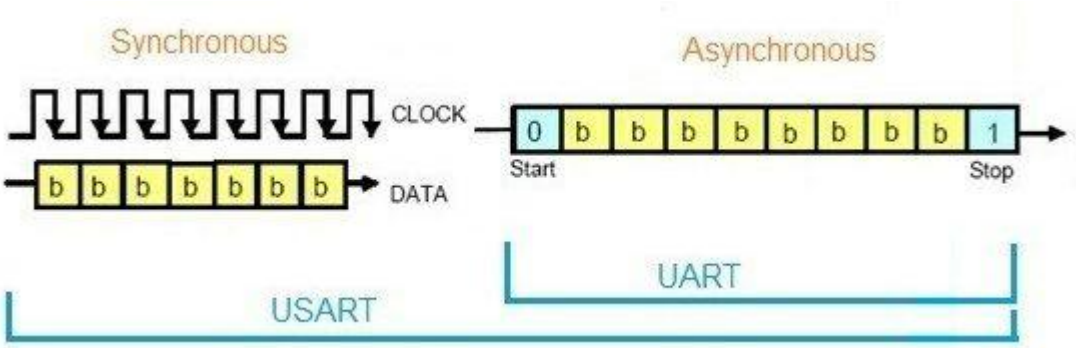
- Simplex: Cho phép giao tiếp một chiều từ máy phát đến máy thu.
- Một nửa song công: liên quan đến việc truyền tải và tiếp nhận xen kẽ.
- Toàn bộ song công: Cho phép truyền và tiếp nhận đồng thời.
- Cấu trúc gói dữ liệu UART:

Bắt đầu bit: báo hiệu sự khởi đầu của một gói mới.

- Khung dữ liệu: Chứa các bit dữ liệu thực tế, thường dao động từ 5 đến 9 bit.
- Bit Parity: Tùy chọn để kiểm tra lỗi.
- Bit dừng: Cho biết phần cuối của gói dữ liệu.
- Tỷ lệ baud: Tốc độ Baud biểu thị tốc độ truyền dữ liệu tính bằng bit-per-second (bps). Tốc độ baud tiêu chuẩn phổ biến bao gồm 9600, 1200, 2400, 4800, 19200, 38400, 57600 và 115200 bps.

Sự khác nhau giữa UART và USART:

UART và USART là hai thành phần thiết yếu trong giao tiếp nối tiếp, mỗi thành phần có đặc điểm riêng: UART: là viết tắt của máy thu/máy phát không đồng bộ phổ quát. Nó hoạt động như một thực thể độc lập, phù hợp cho các ứng dụng tốc độ thấp.



Hình 2.9: Khung truyền của UART

UART sử dụng một chiếc đồng hồ được tạo ra cục bộ, làm cho nó tiết kiệm năng lượng. USART: biểu thị máy thu/máy phát đồng bộ/không đồng bộ toàn cầu. Không giống như UART, USART hỗ trợ các giao thức khác nhau, cho phép giao tiếp tốc độ cao. Nó sử dụng đồng hồ do máy phát tạo ra nhưng có xu hướng tiêu thụ nhiều năng lượng hơn so với các thiết lập UART.

UART đối với vi điều khiển PIC:

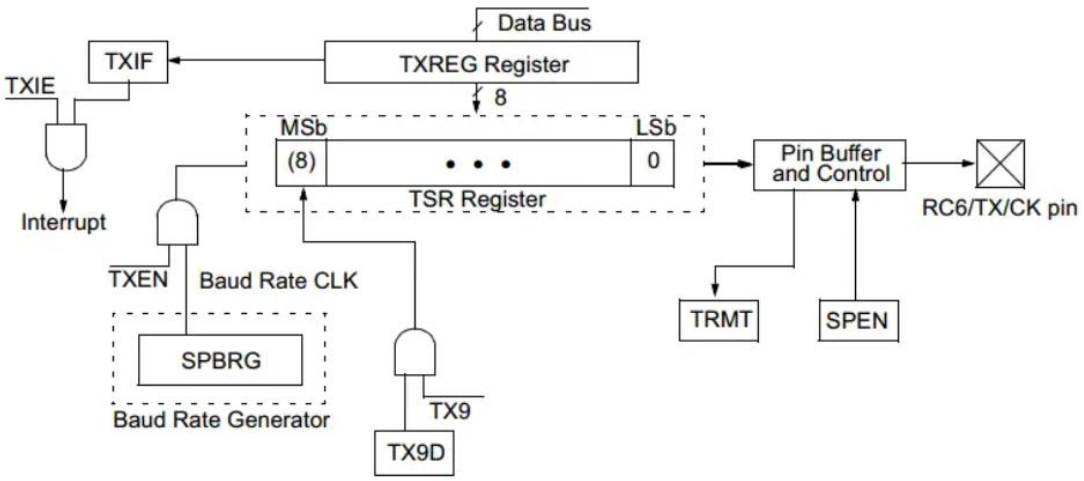
PIC16F877 UART là một thành phần thiết yếu để giao tiếp nối tiếp, truyền và nhận dữ liệu hiệu quả trong khi ưu tiên LSB (bit ít có ý nghĩa nhất) trước tiên. Hệ thống độc lập phần cứng này đảm bảo hoạt động liền mạch bằng cách tuân thủ các định dạng dữ liệu nhất quán và tốc độ baud. Tại trung tâm của chức năng UART PIC16F877 là trình tạo tốc độ baud, có thể định cấu hình cho tỷ lệ dịch chuyển bit x16 hoặc x64 dựa trên bit BRGH. Mặc dù hỗ trợ phần cứng cho chẵn lẻ là không có, việc triển khai phần mềm bit dữ liệu thứ chín bù cho giới hạn này. Đáng chú ý, chế độ không đồng bộ ngừng hoạt động trong chế độ ngủ, đảm bảo hiệu quả năng lượng.

Mô -đun không đồng bộ UART bao gồm các thành phần quan trọng:

- Trình tạo tốc độ baud
- Mạch lấy mẫu
- Máy phát không đồng bộ
- Máy thu không đồng bộ

Chức năng cốt lõi của bộ phát, xoay quanh thanh ghi dịch chuyển (TSR) của bộ truyền (TSR), thu được dữ liệu từ bộ đệm truyền/ghi (TXREG). Can thiệp phần mềm là cần thiết để tải dữ liệu vào thanh ghi TXREG, đảm bảo quá trình truyền dữ liệu trơn tru. Sau khi hoàn thành bit dừng truyền dẫn trước đó, thanh ghi TSR tải khung dữ liệu

mới từ TXREG, tạo điều kiện truyền dữ liệu liên tục. Đáng chú ý, bit cờ TXIF biểu thị một điều kiện ngắt, có thể điều chỉnh thông qua bit TXIE, trong khi bit trạng thái TRMT biểu thị độ trống của thanh ghi TSR, cần phải bỏ phiếu thủ công.



Hình 2.10: Cấu trúc bộ xử lí UART

Sự khởi đầu truyền dẫn đòi hỏi phải đặt bit kích hoạt TXEN, được kết hợp với tải dữ liệu vào thanh ghi TXREG và tạo đồng hồ bằng trình tạo tốc độ baud. Tính linh hoạt tồn tại để bắt đầu truyền bằng cách tải TXREG đầu tiên hoặc bằng cách tải TxREG đồng thời và cho phép TXEN. Vô hiệu hóa TXEN Mid Transmission phá thai quá trình, đặt lại máy phát và hoàn nguyên chân TX/RC6 về trạng thái biến đổi cao. Các bước cấu hình máy phát UART:

- Định cấu hình tốc độ baud: Khởi tạo thanh ghi SPBRG để đạt được tốc độ baud mong muốn, sử dụng bit BRGH cho hoạt động tốc độ cao.

- Kích hoạt cổng nối tiếp không đồng bộ: Kích hoạt cổng nối tiếp không đồng bộ bằng cách định cấu hình các bit đồng bộ và spen.

- Đặt hướng dữ liệu pin: Xác định hướng dữ liệu cho các chân RX và TX (RC6/TX/CK và RC7/RX/DT) cho hoạt động UART.

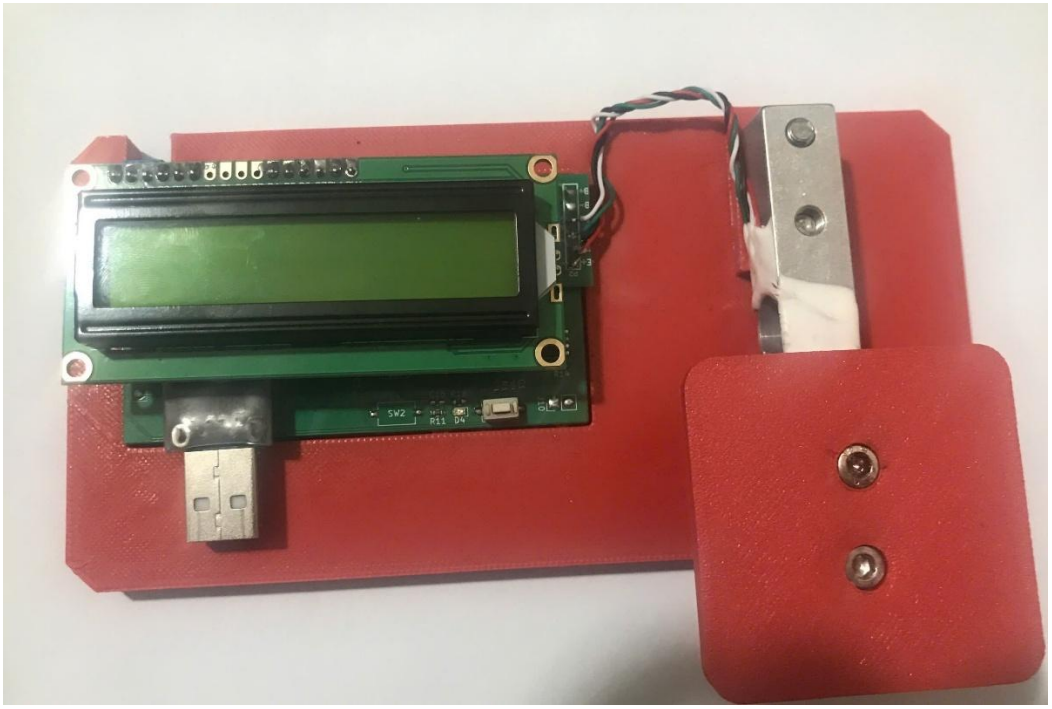
- Bật truyền UART: Kích hoạt truyền UART bằng cách đặt bit TXEN.

Giao tiếp với máy tính thông qua CH340C trên mạch TTL mua sẵn

Hình 2.11: Mạch chuyển TTL sử dụng chip CH340

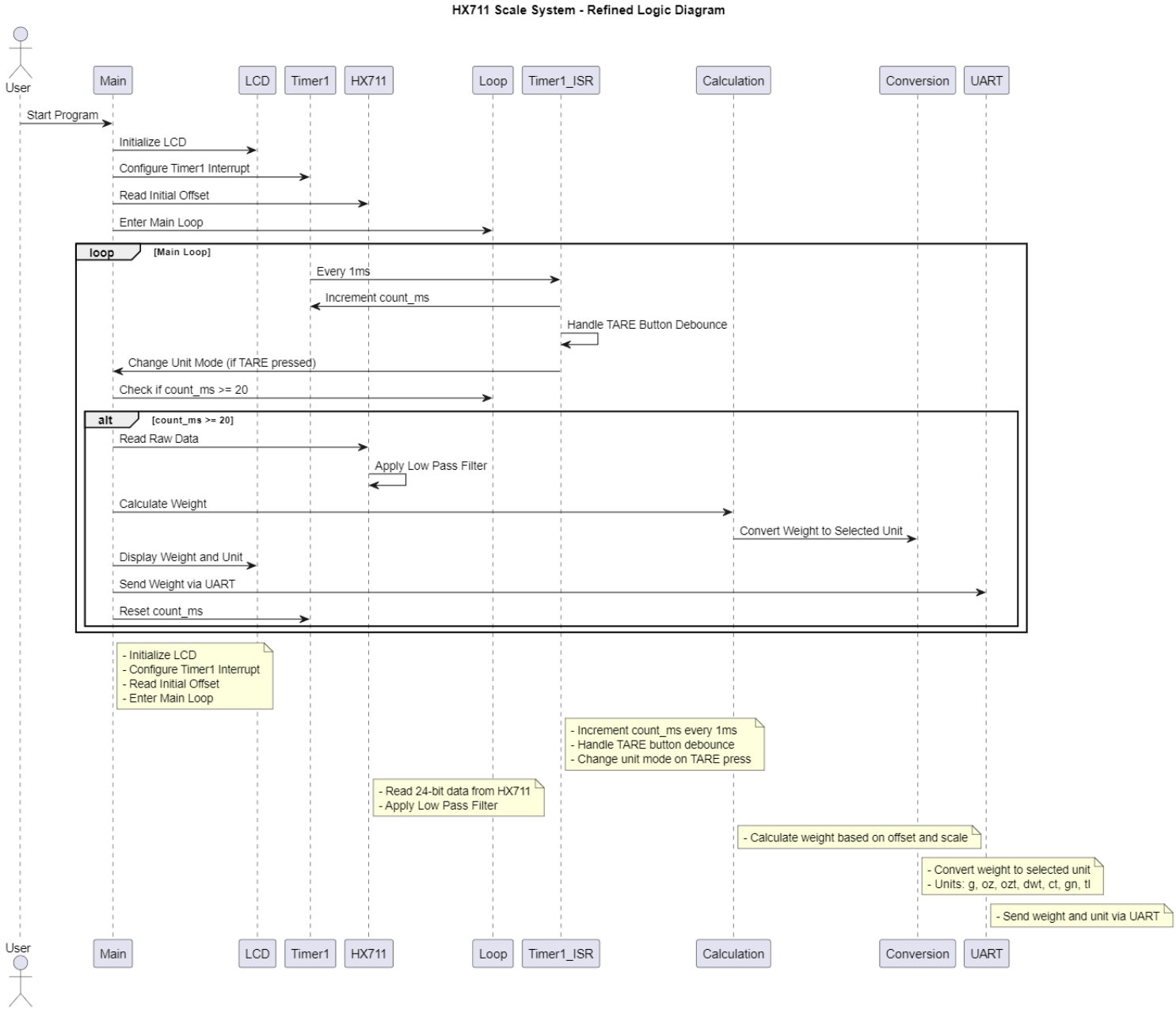
CHƯƠNG 3: KẾT QUẢ THỰC HIỆN VÀ KẾT LUẬN

3.1. Kết quả thực hiện



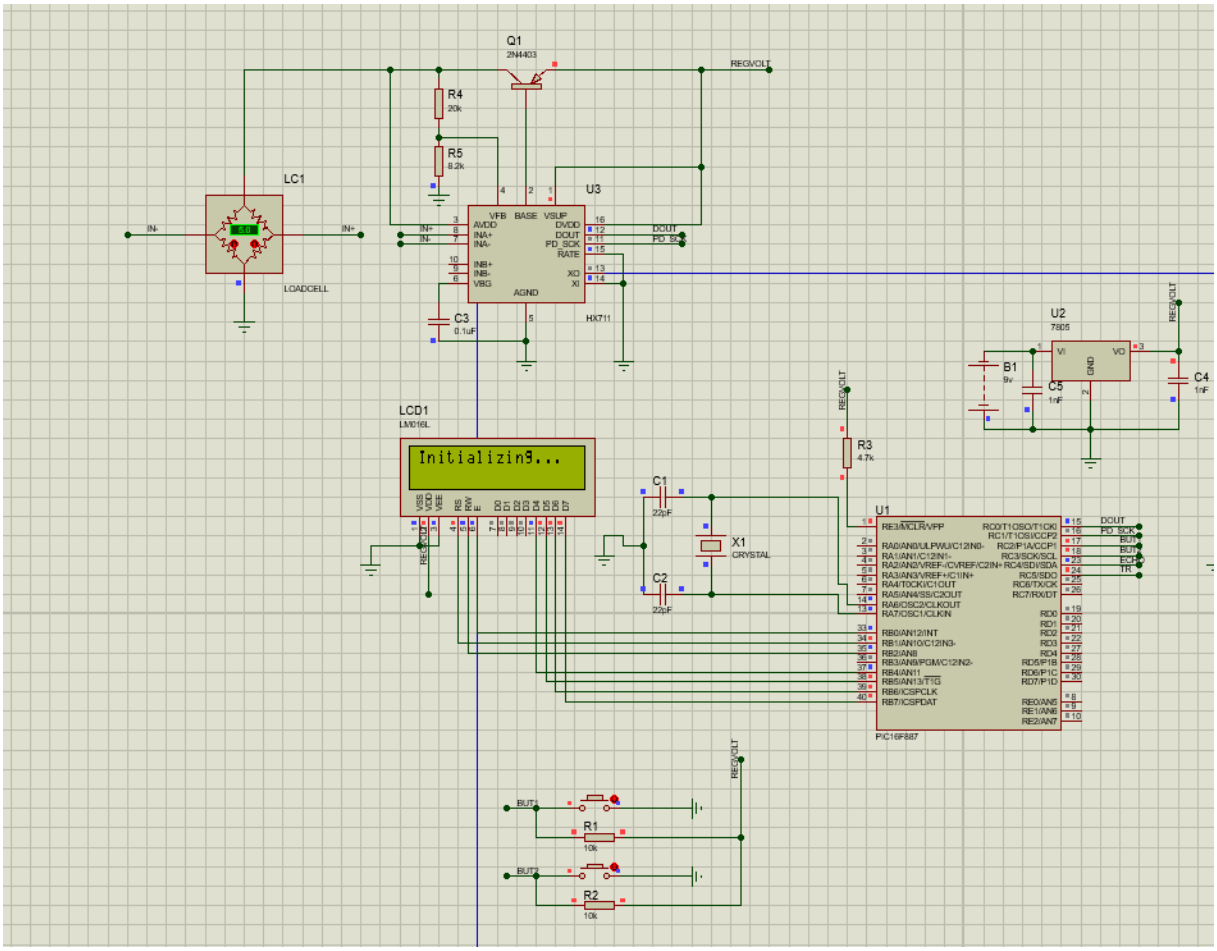
Hình 3.1: Sản phẩm thực tế

3.1.1. Lập trình hệ thống



Hình 3.2: Giải thuật và thứ tự thực thi.

3.1.2. Kết quả mô phỏng

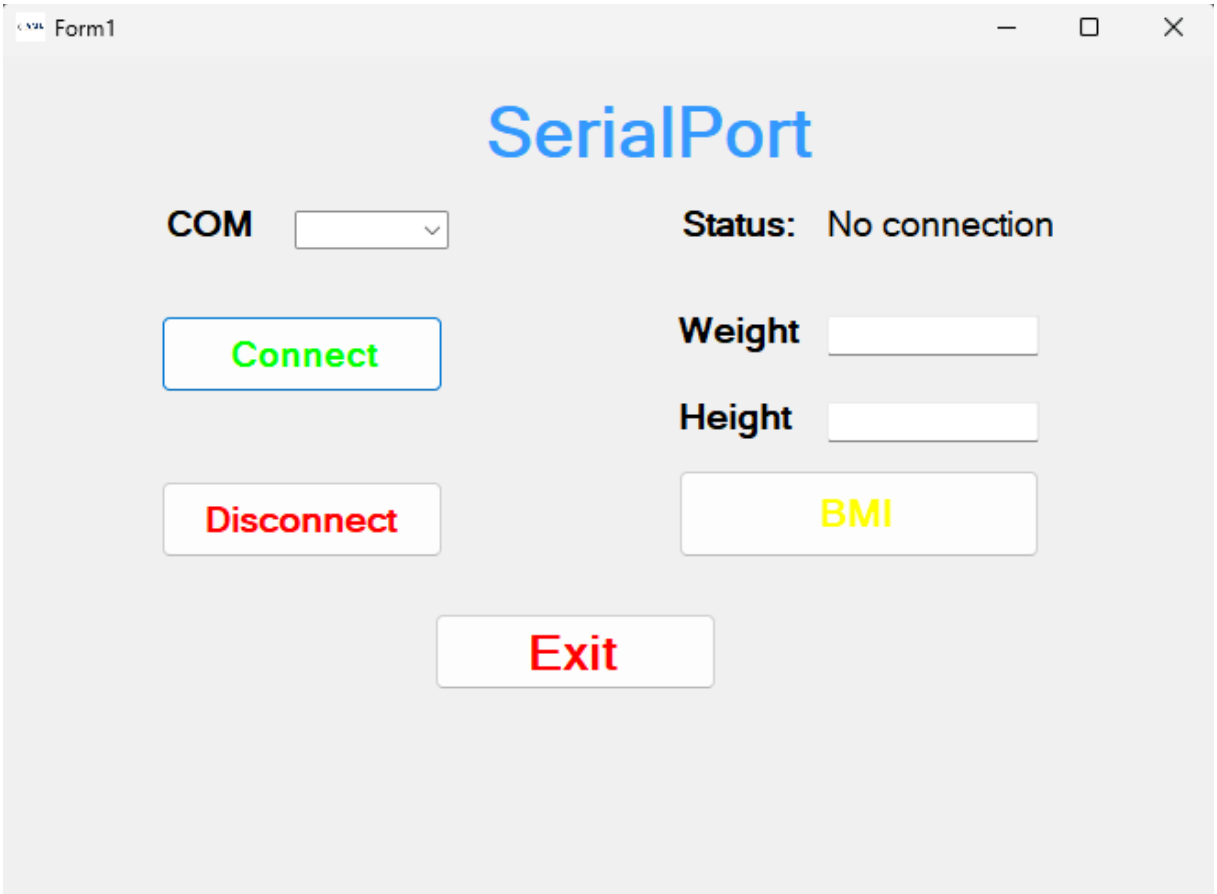


Hình 3.3: Kết quả mô phỏng

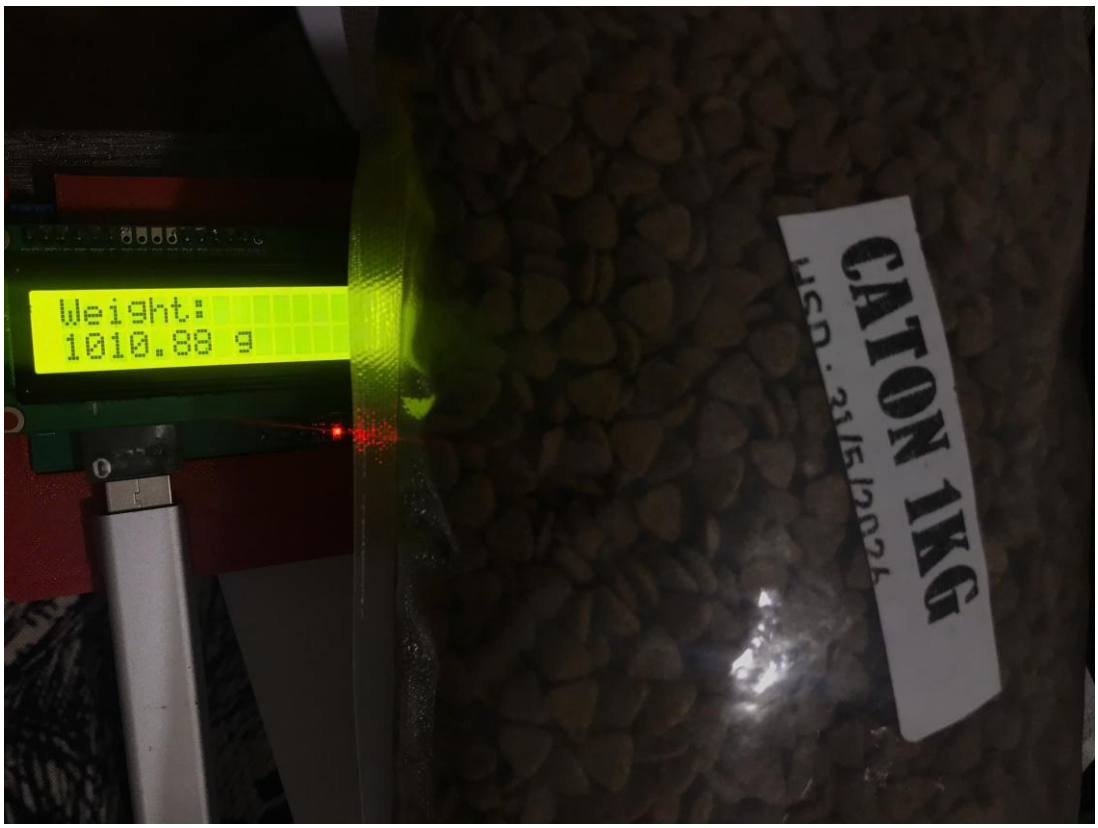
3.1.3. Kết quả thực nghiệm

Thông số cấu hình	
Điện áp hoạt động	6V-25V
Khối lượng đo	0-5000g
Ngoại vi	LCD, nút nhấn, Loadcell, Modul HX711
Sai số	≤ 0,5%
Các chế độ đo	- g (gram - gam): Đây là đơn vị đo khối lượng cơ bản trong hệ mét. - oz (ounce - aoxơ): Đây là đơn vị đo khối lượng trong hệ đo lường Anh-Mỹ - ozt (troy ounce - aoxơ troy): Đây cũng là đơn vị đo khối lượng được sử dụng trong ngành kim hoàn - ct (carat - carat): Đây là đơn vị đo khối lượng được sử dụng cho đá quý

	- gn (grain - grên): Đây là đơn vị đo khối lượng cổ xưa, hiện nay ít được sử dụng trong đo lường thông thường. - tl (tola - tola): Đây là đơn vị đo khối lượng truyền thống được sử dụng ở Nam Á



Hình 3.4: GUID bằng C#



Hình 3.5: Cân bao thức ăn cho mèo

Ở đây chúng em thực hiện đo với vật đo là bao thức ăn cho mèo còn nguyên tem khối lượng 1kg, thêm phần vỏ bao bì khoảng 10g, tất cả là 1010g. Sau khi đo 15 lần thu được kết quả trình bày ở 2 bảng dưới đây như sau:

Bảng 3.1: Bảng sai số giữa hai lần đo liên tiếp

Lần đo	**Giá trị đo (g)**	**Sai số tuyệt đối so với lần trước (g)**	**Sai số tương đối so với lần trước (%)**
1	1009.25	-	-
2	1010.88	1.63	0.16
3	1008.50	2.38	0.24
4	1011.20	2.70	0.27
5	1009.80	1.40	0.14
6	1010.55	0.75	0.07
7	1008.90	1.65	0.16
8	1011.00	2.10	0.21
9	1010.10	0.90	0.09
10	1009.50	0.60	0.06
11	1011.45	1.95	0.19

Lần đo	Giá trị đo (g)	Sai số tuyệt đối so với lần trước (g)	Sai số tương đối so với lần trước (%)
12	1008.17	3.28	0.32
13	1010.30	2.13	0.21
14	1009.75	0.55	0.05
15	1011.30	1.55	0.15

Nhận xét:

- Sai số giữa hai lần đo dao động từ 0.05% đến 0.32%.
- Lần đo 12 có sự thay đổi lớn nhất so với lần trước đó (3.28 g, 0.32%).

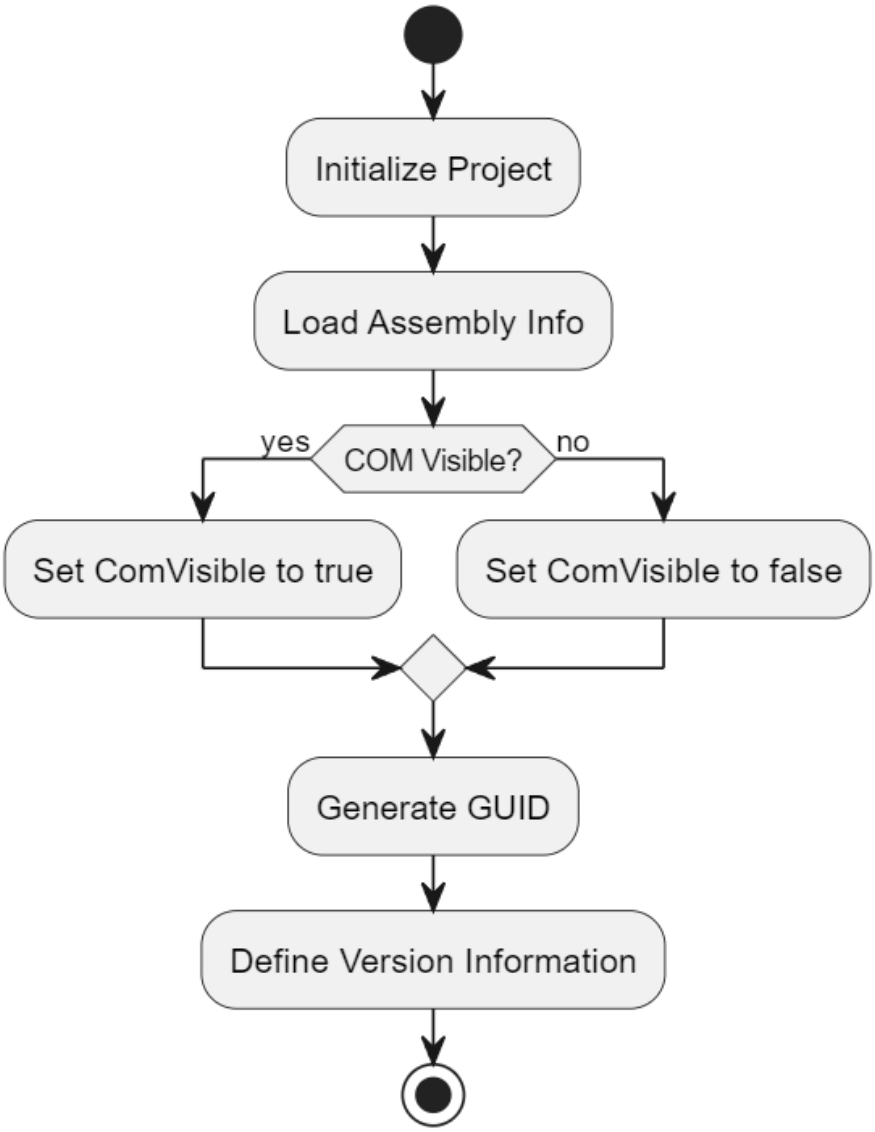
Bảng 3.2: Bảng sai số so với giá trị thực (1010 g)

Lần đo	Giá trị đo (g)	Sai số tuyệt đối so với 1010 g (g)	Sai số tương đối so với 1010 g (%)
1	1009.25	0.75	0.07
2	1010.88	0.88	0.09
3	1008.50	1.50	0.15
4	1011.20	1.20	0.12
5	1009.80	0.20	0.02
6	1010.55	0.55	0.05
7	1008.90	1.10	0.11
8	1011.00	1.00	0.10
9	1010.10	0.10	0.01
10	1009.50	0.50	0.05
11	1011.45	1.45	0.14
12	1008.17	1.83	0.18
13	1010.30	0.30	0.03
14	1009.75	0.25	0.02
15	1011.30	1.30	0.13

Nhận xét:

- Lần đo 9 có sai số nhỏ nhất so với giá trị thực (0.10 g, 0.01%).
- Lần đo 12 có sai số lớn nhất (1.83 g, 0.18%).
- Sai số trung bình so với giá trị thực là nhỏ hơn 0.2%, cho thấy độ chính xác của phép đo khá cao.

3.1.4. Lập trình giao diện bên ngoài



Hình 3.6: Flowchart GUID

3.2. Kết luận và hướng phát triển

3.2.1. Ưu điểm (Loadcell và HX711)

Độ chính xác cao:

- Loadcell kết hợp với HX711 có khả năng đo trọng lượng với độ chính xác rất cao,sai số rất nhỏ.

- HX711 là một bộ chuyển đổi tín hiệu tương tự sang số (ADC) 24-bit chuyên dụng cho loadcell, giúp giảm thiểu sai số và tăng độ phân giải.

Đo trọng lượng trực tiếp:

- Loadcell đo trực tiếp trọng lượng, phù hợp cho các ứng dụng cần xác định khối lượng vật thể.

Tính ổn định và độ bền cao:

- Loadcell thường có cấu trúc chắc chắn, chịu được tải trọng lớn và hoạt động ổn định trong nhiều điều kiện môi trường.

Khả năng đo lường đa dạng:

- Loadcell có nhiều loại với các dải đo khác nhau, phù hợp cho các ứng dụng từ cân tiểu ly đến cân công nghiệp.

Chi phí hợp lý:

- HX711 có giá thành rẻ.
- Loadcell có nhiều loại giá từ thấp đến cao, tùy vào nhu cầu sử dụng.

3.3.2. Nhược điểm (Loadcell và HX711)

Yêu cầu hiệu chuẩn:

- Loadcell cần được hiệu chuẩn định kỳ để đảm bảo độ chính xác.
- Môi trường có nhiệt độ và độ ẩm thay đổi sẽ ảnh hưởng đến độ chính xác của Loadcell.

Ảnh hưởng bởi nhiễu điện:

- Tín hiệu từ loadcell có thể bị ảnh hưởng bởi nhiễu điện từ, đặc biệt trong môi trường công nghiệp.
- Cần có biện pháp chống nhiễu phù hợp để đảm bảo độ chính xác.

Giới hạn về tải trọng:

- Mỗi loadcell có một giới hạn tải trọng nhất định. Việc vượt quá giới hạn này có thể gây hỏng hóc hoặc làm giảm độ chính xác.

Tóm lại, loadcell và HX711 cung cấp giải pháp đo trọng lượng chính xác và ổn định, nhưng cần lưu ý đến các yếu tố như hiệu chuẩn, nhiễu điện và giới hạn tải trọng.

3.3.2. Kết luận

Mạch phần cứng:

- Đã được thiết kế và lắp ráp ổn định, đảm bảo tính chính xác cao trong việc đo trọng lượng và độ bền của hệ thống.
- Sử dụng Loadcell và HX711 được chọn lựa kỹ càng để giảm thiểu sai số và nhiễu tín hiệu.

Vận hành:

- Hệ thống vận hành tốt, đáp ứng nhanh với các thay đổi trọng lượng và duy trì sự ổn định của kết quả đo.
- Quá trình hiệu chuẩn được thực hiện kỹ lưỡng để đảm bảo kết quả chính xác trong quá trình vận hành.

Hướng phát triển:

1. Tích hợp thêm các ngoại vi:

- Tích hợp màn hình hiển thị để hiển thị kết quả đo trọng lượng trực tiếp.
- Thêm đèn cảnh báo hoặc còi báo động khi trọng lượng vượt quá ngưỡng cho phép.
- Tích hợp động cơ để tự động hóa quá trình cân hoặc phân loại sản phẩm.
- Tích hợp thêm các giao thức kết nối không dây như wifi, bluetooth để truyền tải thông tin cân lên các hệ thống giám sát.

2. Nâng cấp phần cứng:

- Sử dụng loadcell có độ chính xác và dải đo rộng hơn để phù hợp với nhiều ứng dụng khác nhau.
- Nâng cấp vi điều khiển để tăng khả năng xử lý tín hiệu và tích hợp thêm các tính năng thông minh.
- Tối ưu hóa thiết kế mạch để giảm nhiễu điện từ và tăng độ ổn định của hệ thống.
- Tối ưu hóa thiết kế mạch để giảm kích thước và tăng tính linh hoạt trong việc tích hợp và các hệ thống khác.

3. Phát triển phần mềm hỗ trợ:

- Xây dựng giao diện người dùng thân thiện trên máy tính hoặc thiết bị di động để dễ dàng theo dõi và điều khiển hệ thống cân.
- Phát triển phần mềm phân tích dữ liệu để thống kê và báo cáo kết quả cân.
- Xây dựng hệ thống hiệu chuẩn tự động.

TÀI LIỆU THAM KHẢO

[1] A. De Marcellis, G. Ferri, and P. Mantenuto, “Resistive Sensor Interfacing,” in *Giant Magnetoresistance (GMR) Sensors: From Basis to State-of-the-Art Applications*, C. Reig, S. Cardoso, and S. C. Mukhopadhyay, Eds. Berlin, Heidelberg: Springer, 2013, pp. 71–102. doi: 10.1007/978-3-642-37172-1_4.

[2] S. Shirotori, A. Kikitsu, Y. Higashi, Y. Kurosaki, and H. Iwasaki, “Symmetric Response Magnetoresistance Sensor With Low 1/f Noise by Using an Antiphase AC Modulation Bridge,” *IEEE Trans. Magn.*, vol. 57, no. 2, pp. 1–5, Feb. 2021, doi: 10.1109/TMAG.2020.3012655.

[3] Loadcell's datasheet: https://electronperdido.com/wp-content/uploads/2015/09/Celda-de-carga-CZL635-datasheet.pdf

[4] Pic’s datasheet: https://ww1.microchip.com/downloads/en/devicedoc/41291d.pdf

[5] HX711’s datasheet:

https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

[6] LCD-1602A :https://www.alldatasheet.com/datasheet-pdf/download/1574132/CRYSTAIFONTZ/LCD-1602A.html

[7] CCS C: https://www.ccsinfo.com/downloads/ccs_c_manual.pdf

[8] PICkit3: https://ww1.microchip.com/downloads/en/devicedoc/51795b.pdf

[9] Program: https://gist.github.com/schappim/415955a224824ab18f93

Link Github chứa tư liệu, thiết kế, demo project: