THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

PBL 3: DESIGNING EMBEDDED SYSTEMS WITH MICROCONTROLLERS AND SENSORS

PROJECT: DIGITAL WEIGHING SCALE FOR SELLING GOODS

Lecturers: Ph.D DO THE CAN

Student: VO TAN PHU

PHAN HUU THANG

Student ID Number: 101200281

101200242

Class: 20CDT2 & 20CDT1

Danang, May 2023

DA NANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM
Independent – Freedom - Happiness

MISSION OF PROJECT PBL3

Num	Name	ID	Class	Industry
1	Vo Tan Phu	101200281	20CDT2	Mechatronic Engineering
2	Phan Huu Thang	101200242	20CDT1	Mechatronic Engineering

1. Lecturer: PhD. Do The Can, PhD. Dang Phuoc Vinh

Name of project: Digital Weighing Scale For Selling Goods

2.The subject is subject to: ☐ Having signed an agreement on intellectual property for the results of implementation

3. Content of the explanations and caculations:

a) General part:

Num	Name	Content
1	Vo Tan Phu	- Get an overview about Digital Weighing Scale For Selling Goods
2	Phan Huu Thang	- Hardware and software design caculation - Prepare presentation, slides and drawings as required

b) Private part:

Num	Name	Content
1	Vo Tan Phu	- Calculation and design of electronic and control parts - Design HMI interface to monitor system operation
2	Phan Huu Thang	- Calculation of mechanical design - System control programming

4. Drawing, graphs (specify types and sizes of drawings):

a) General part:

Num	Name	Content
1 2	Vo Tan Phu Phan Huu Thang	- Make the overall drawing of the machine (1 – A0)

b) Private part:

Num	Name	Content
1	Vo Tan Phu	- Make an algorithm flowchart drawing
2	Phan Huu Thang	- Make a drawing of an electrical circuit diagram

5. Project assignment date: 11/2/2023

6. Project completion date: 8/6/2023

DA NANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM
Independent – Freedom - Happiness

PBL3 PROJECT IMPLEMENTATION SCHEDULE

Time	Content
Week 1 (7/1 – 14/1)	Meet with teacher to discuss the topic
Week 2 (14/1 – 21/1)	Find out about the project
Week 3 (21/1 – 4/2)	Learn about different types of sensors and get ideas for projects
Week 4 (4/2 – 11/2)	Project report to teacher
Week 5 (11/2 – 18/2)	Draw schematic diagrams, 3D printed circuits and models for the system.
Week 6 (18/2 – 25/2)	Report the overall drawing of the model to the drawing A0. Report 3D printed circuit diagram, export layout, report product cost.
Week 7 – 12 (25/2 – 8/4)	Project programming
Week 12 – 16 (8/4 – 6/5)	Programming interfaces and project reports.
Week 16 (6/5 – 14/5)	Finalize the project, overall report.
Week 17 (14/5 – 21/5)	Complete project
Week 18 – 20 (21/5 -8/6)	Prepare and protect the project

CONTENT

INTRODUCTION

In the modern industrialization and modernization of the country, the mechanical industry has become a spearhead industry and plays a crucial role in various other industries.

As students of the Mechanical Engineering department at the University of Engineering and Technology, after three years of studying and training under the dedicated guidance of our professors, we have gained valuable knowledge and skills. In line with the assigned task of designing a device using microcontrollers and sensors, we have unanimously decided to undertake the project of Digital weghing sacle for selling goods. This project is driven by the essential nature of people's lives, where the need for efficient purchase and quick payment methods is increasing to meet the growing demands of society.

During the course of this project, we acknowledge that there may be limitations in terms of time constraints and specialized knowledge. Additionally, as this is our first endeavor in electronic design, it is inevitable that there may be unintended mistakes and shortcomings. We highly appreciate and eagerly look forward to receiving sincere feedback and guidance from you, esteemed professors, to help us successfully complete this design project.

We would like to express our heartfelt gratitude to all the professors in the Mechanical Engineering department, especially to Professor Đo The Can, for their unwavering support and guidance throughout the process of completing this project.

Danang, March 23, 2023
Author

VO TAN PHU

CHAPTER 1 INTRODUCTION TO DIGITAL WEIGHING SCALE AND SALES MANAGEMENT

1.1 Problem Statement

- In today's digital and electronic industry advancements, automation is becoming increasingly prevalent. With the development of microprocessors and digital circuits, information processing has become faster, serving the needs of human life.
- In the retail industry, including large agricultural supermarkets and small grocery stores, the demand for consumer goods is constantly increasing to meet the growing needs of society. However, this development also poses challenges in the sales process, particularly in the checkout process where customers often wait in line to weigh their products and make payments, resulting in reduced efficiency. The time-consuming process of weighing and payment can hinder the overall effectiveness of sales. To address these issues, we have decided to design an Electronic Weighing and Sales Management System to facilitate accurate, easy, and efficient payment and sales information management, thereby increasing productivity and delivering high value to users.
- The system utilizes a PIC16F877A microcontroller and communicates with a computer using QR code technology to identify and quantify the items for payment. For software development, we employ CCS for microcontroller programming and Visual Studio 2022 with C# programming language to develop the software component of the system.

1.2 Objectives

- Design and develop an electronic weighing scale and sales management system with the following functions:
  - + Measure weight using a load cell connected to an HX711 module to convert analog data into digital data and amplify the signal.
  - + Include functional buttons with 4 modes: a reset button to zero the weight value, a hold button to retain the weight value, a counting button to count the number of weighings, and a unit conversion button (kg, g, oz).
  - + Displaying the weight value and modes on a 16x2 LCD screen.
  - + The sales management system will process QR codes to read the code and retrieve the weight value from the electronic scale for price calculation.
  - + Print a sales receipt after the scanning process is completed.
  - + Include functions to add new products, delete and update items, and other information in the system.

1.3 Limitations

- o Capacity: The maximum weight capacity of the scale is 10 kg, and it operates within a range of 18 g to 10 kg.
- o Stable QR code recognition capability.

- Objects should be placed directly on the weighing surface without any overhang to ensure accurate weighing.

1.4 Product Images of the Model

Figure 1.1 Actual model of the product

CHAPTER 2: INTRODUCTION TO HARDWARE COMPONENTS

2.1 Overview of PIC16F877A

- The PIC16F887A microcontroller is a member of the PIC16F family, manufactured by Microchip. It features a 14-bit instruction set with 35 instructions. This chip is widely used in Vietnam as well as globally due to its comprehensive functionality typical of microcontrollers and its suitability for basic applications.
- The PIC16F877A is produced and packaged in two types: PDIP (Plastic Dual Inline Package) and TQFP (Thin Quad Flat Package). The choice of packaging type depends on the specific requirements of the applications users are working on.

Figure 2.1 Actual image of PIC 16F877A microcontroller

2.1.1 General structure of the PIC 16F877A microcontroller is as follows:

- An 8/16-bit CPU built on a modified Harvard architecture.
- Optional Flash and ROM memory ranging from 256 bytes to 256Kbytes.
- Synchronous/asynchronous serial communication peripherals including AUSART, USART and EUSARTs
- Input/output ports with logic levels ranging from 0V to 5.5V, corresponding to logic 0 and logic 1.
- Capable of operating at various oscillator frequencies (refer to the diagram).
- An ADC converter with 10/12-bit resolution.
- Voltage Comparators.
- Two CCP (Capture/Compare/PWM) modules.
- Belongs to the PIC 16F87xxx family with a 14-bit instruction set consisting of 35 instructions.
- All instructions take one machine cycle, except for subroutine calls which take two machine cycles.
- One instruction cycle of the microcontroller consists of 4 clock cycles. If a 4 MHz crystal is used, the instruction cycle frequency will be 1 MHz, corresponding to a cycle time of 1μs.

- Flash program memory with a capacity of 8K x 14 bits, capable of being written/erased up to 100,000 times.
- RAM data memory with a capacity of 368 bytes.
- EEPROM memory with a capacity of 256 bytes, capable of being written/erased up to 1,000,000 times and can store data for over 40 years.
- Operates with a power supply of 4.0 ÷ 5.5 VDC (refer to the diagram).
- Includes a Sleep mode to conserve power.
- Provides 5 I/O ports (named A, B, C, D, E) with 3 pins each.
- Supports USB, Ethernet, CAN, LIN, IrDA communication interfaces.

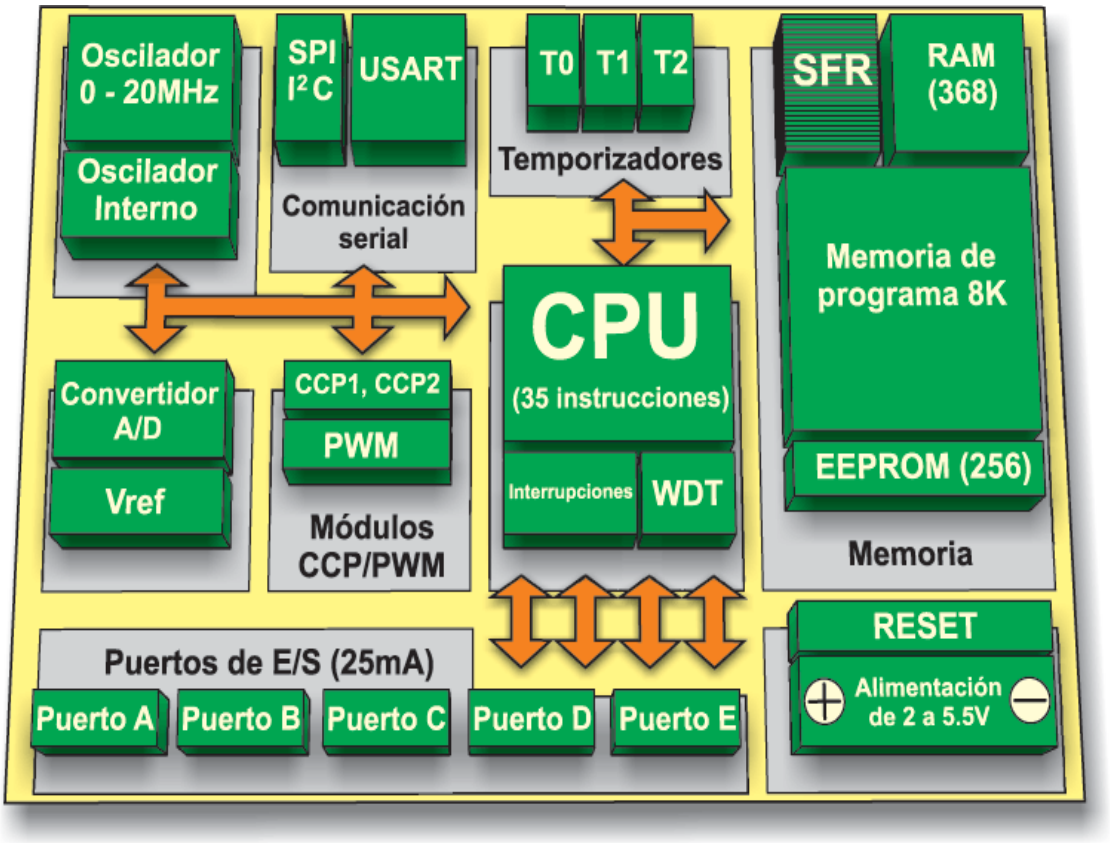


Figure 2.2 General structure of the PIC 16F877A

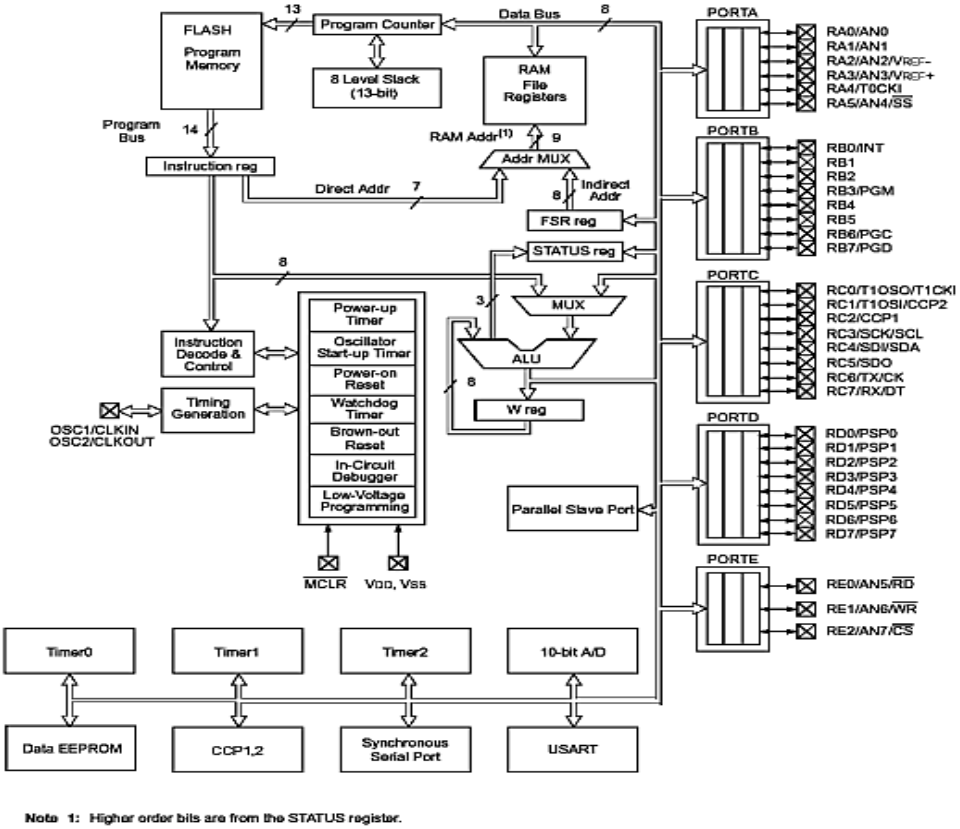
- The peripheral features of the PIC16F877A microcontroller include the following functional blocks:

- Timer 0: An 8-bit timer with an 8-bit prescaler.
- Timer 1: A 16-bit timer with a prescaler. It can also function as an external clock input for counting even in sleep mode.
- Timer 2: An 8-bit timer with a prescaler and a postscaler. It also has a Capture/Compare/PWM module for generating variable pulse width signals.
- Serial communication standards including SSP (Synchronous Serial Port), SPI, and I2C. USART (Universal Synchronous/Asynchronous Receiver/Transmitter) for serial communication with 9-bit addressing.

- Parallel Slave Port (PSP) for parallel communication with external control pins RD, WR, CS.

2.1.2 Block diagram and pin functions:

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

Figure 2.3 Block diagram of the PIC 16F877A

- Pins diagram:

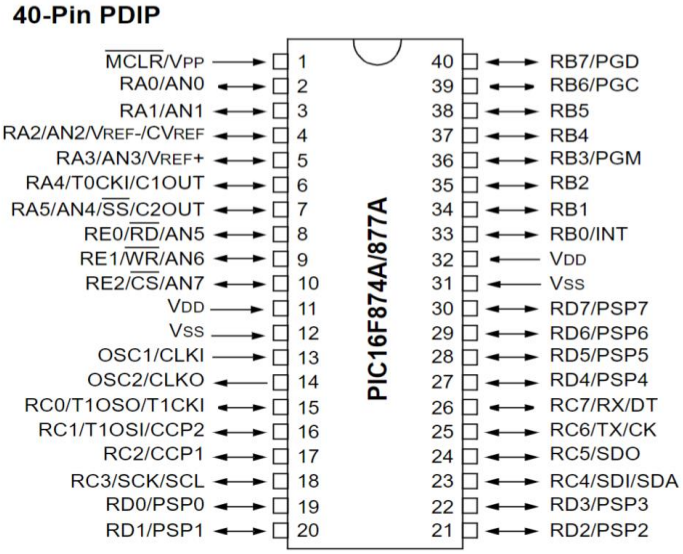


Figure 2.4 Pins diagram of the PIC 16F877A

2.1.3 Memory:

- EEPROM Memory: The PIC16F877A microcontroller has an integrated EEPROM (Electrically Erasable Programmable Read-Only Memory) data memory with a capacity of 256 bytes. It is considered as a special data memory device connected to the data bus. The EEPROM memory can be read from and written to under program control. It is commonly used to store non-volatile data such as standard constants, user data, and retains its data even when power is disconnected.
- Data Memory: The data memory is divided into 4 banks, with each bank having a capacity of 128 bytes of static RAM. Each bank consists of special function register (SFR) registers located in the low address space and general-purpose register (GPR) registers located in the high address space. SFR registers, such as STATUS, INTCON, and FSR, are frequently used and are allocated in all banks to facilitate easy access.

2.2 Overview of 16x2 LCD

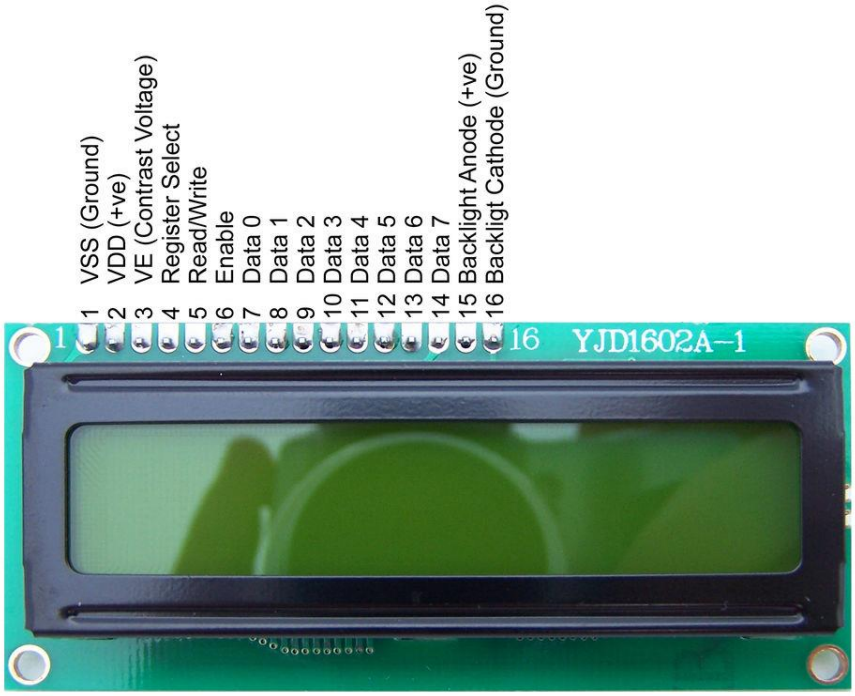


Figure 2.5 Pins diagram of 16x2 LCD

- The LCD (Liquid Crystal Display) is widely used in various applications of microcontrollers. LCDs offer several advantages over other display types. They have the ability to display a wide range of characters, numbers, and graphic symbols in a visually appealing manner. They are easily interfaced with different communication protocols, require minimal system resources, and are cost-effective.
- The function of each pin of the LCD 1602 is as follows:
  - Pin 1 - VSS: Ground pin of the LCD, connected to the GND of the controlling circuit.
  - Pin 2 - VDD: Power supply pin for the LCD, connected to the VCC (5V) of the controlling circuit.

- Pin 3 - VE: Contrast adjustment pin of the LCD.
- Pin 4 - RS: Register select pin, connected to logic "0" or logic "1":
+ Logic "0": Bus DB0 - DB7 is connected to the LCD's instruction register (in "write" mode) or to the LCD's address counter (in "read" mode).
+ Logic "1": Bus DB0 - DB7 is connected to the LCD's data register (DR) internally.
- Pin 5 - R/W: Read/Write mode select pin, connected to logic "0" for write or logic "1" for read.
- Pin 6 - E: Enable pin. After the signals are placed on the bus DB0-DB7, the commands are accepted only when a pulse is applied to this pin as follows:
+ In write mode: The data on the bus is transferred to the LCD's internal register upon detection of a high-to-low transition of the signal on the E pin.
+ In read mode: The data is output from the LCD to the bus DB0-DB7 upon detection of a low-to-high transition on the E pin, and the LCD holds the data on the bus until the E pin goes low.
- Pins 7 to 14 - D0 to D7: These are the 8 data lines used to exchange information with the microprocessor unit (MPU). There are two modes for using these 8 data lines: 8-bit mode (data is transmitted on all 8 lines, with MSB being DB7) and 4-bit mode (data is transmitted on 4 lines from DB4 to DB7, with MSB being DB7).
- Pin 15 - A: Positive power supply for the backlight.
- Pin 16 - K: Ground for the backlight.

2.3 Loadcell and HX711 Signal Amplification Circuit

2.3.1 Loadcell Sensor

a) Concept

- A loadcell is a sensor device used to convert force or weight into electrical signals. Loadcells are typically used to sense large forces, static or slowly varying forces. In some cases, loadcells are designed to measure forces dependent on their specific design.

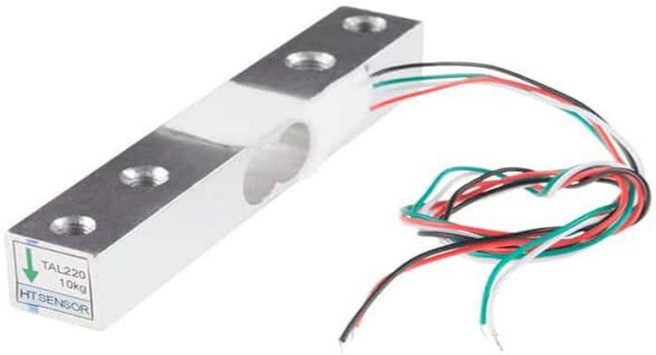


Figure 2.6 10Kg Loadcell Shape

- Technical specifications:
  + Rated load capacity: 10 kg
  + Rated output (mV/V): 2.0 ± 0.15
  + Accuracy class: C2
  + Input resistance: 402 ± 6 Ω
  + Output resistance: 350 ± 3 Ω

b) Structure and Working Principle of Loadcell

- A loadcell consists of two main components: the strain gauge and the load element. The strain gauge is a specialized resistor, usually as small as a fingertip, which changes resistance when compressed or stretched. It is supplied with a stable power source and is bonded to the load element, which is an elastic metal bar that can withstand loads.
- The loadcell operates based on the Wheatstone bridge principle of balanced resistance. The applied force causes a change in the resistance of the strain gauges in the bridge circuit, resulting in an output voltage.

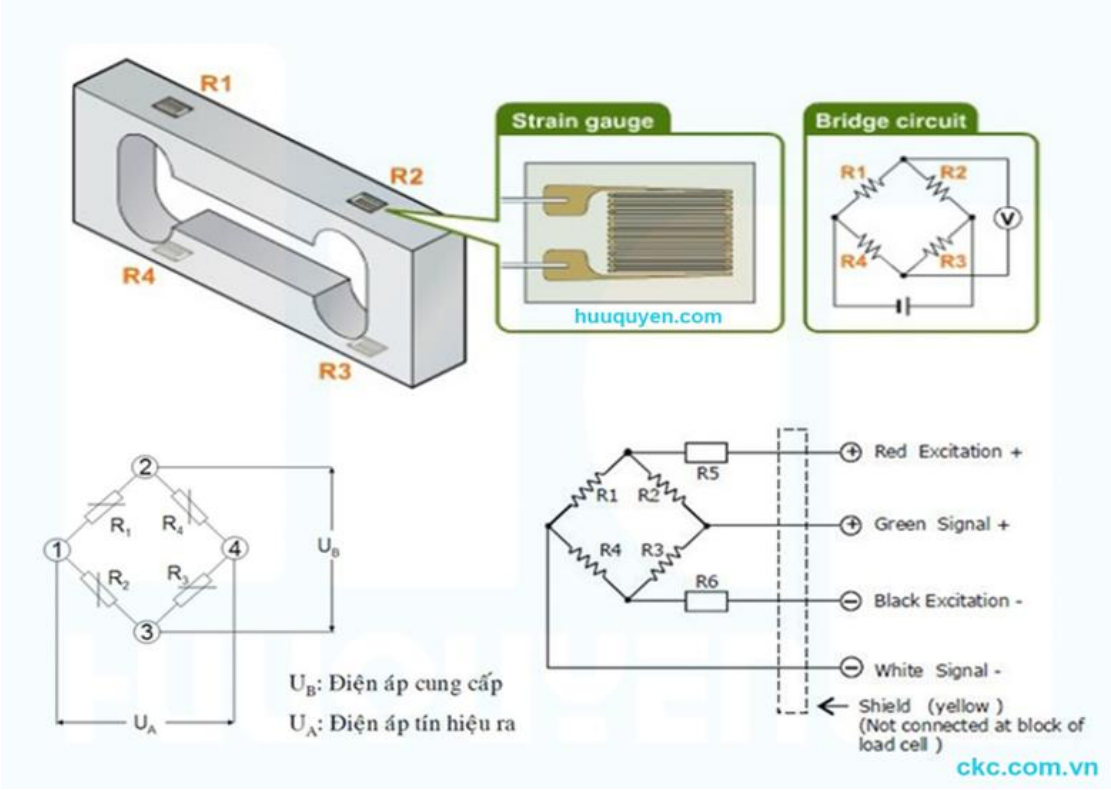


Figure 2.7 Working Principle of Loadcell

* The resistance of the strain gauge can be calculated using the formula:

$$R = \rho.\frac{l}{s}$$

where: ρ: is the resistivity of the material (Ω.m)
l: is the length of the metal piece (m)
s: is the cross-sectional area of the metal piece (m^2)

+ When the material is stretched, the length (l) of the metal piece increases, resulting in an increase in resistance since it is directly proportional to the length (l). At the same time, the cross-sectional area (s) decreases.
+ When the material is compressed, the cross-sectional area (s) of the metal piece increases, resulting in a decrease in resistance since it is inversely proportional to the cross-sectional area (s). Meanwhile, the length (l) decreases.

* By measuring the change in resistance, the loadcell can accurately determine the applied force or weight. This change in resistance is converted into an electrical signal, typically amplified and processed by additional circuitry, such as the HX711 amplifier, to obtain a usable output for measurement and control applications.

- An excitation voltage is applied to the two input terminals of the load cell, and the output signal voltage is measured between the remaining two terminals.
- In the unloaded state, the output voltage signal is close to zero or approximately zero when four resistors are properly connected in terms of value.

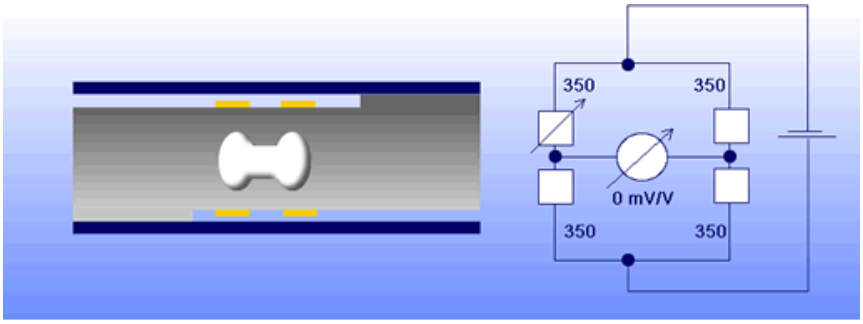


Figure 2.8 Illustrates the operation of a Loadcell in the unloaded state

2.3.2 Module HX711

- The HX711 module is a 24-bit ADC (Analog-to-Digital Converter) designed to amplify signals from load sensors in industrial control applications.

* Structure diagram of the HX711 module

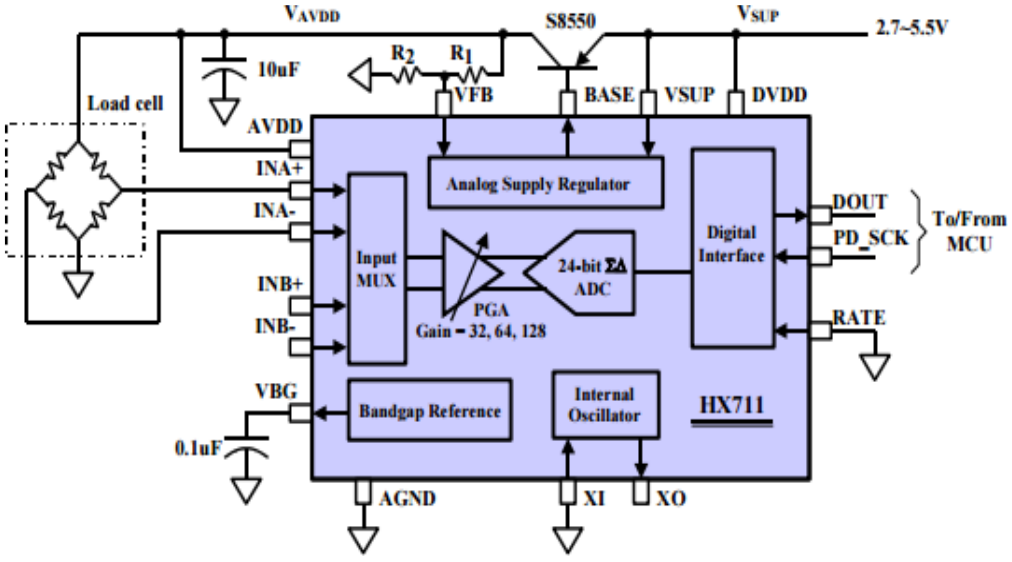


Figure 2.9 Structure diagram HX711

- The HX711 module has two input channels, labeled A and B, and the amplifier can be programmable.
    + Channel A can be programmed with a gain factor of 64 or 128, corresponding to a resolution of ±20mV and ±40mV respectively when supplied with a 5V input at the AVDD pin.
    + Channel B has a fixed gain factor of 32.
- The HX711 communicates with the MCU (Microcontroller Unit) using two wires for data and clock. Its function is to amplify the output signal and convert the analog signal into a digital signal.

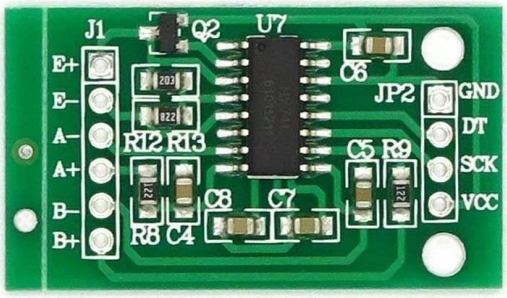


Figure 2.10 HX711 module image

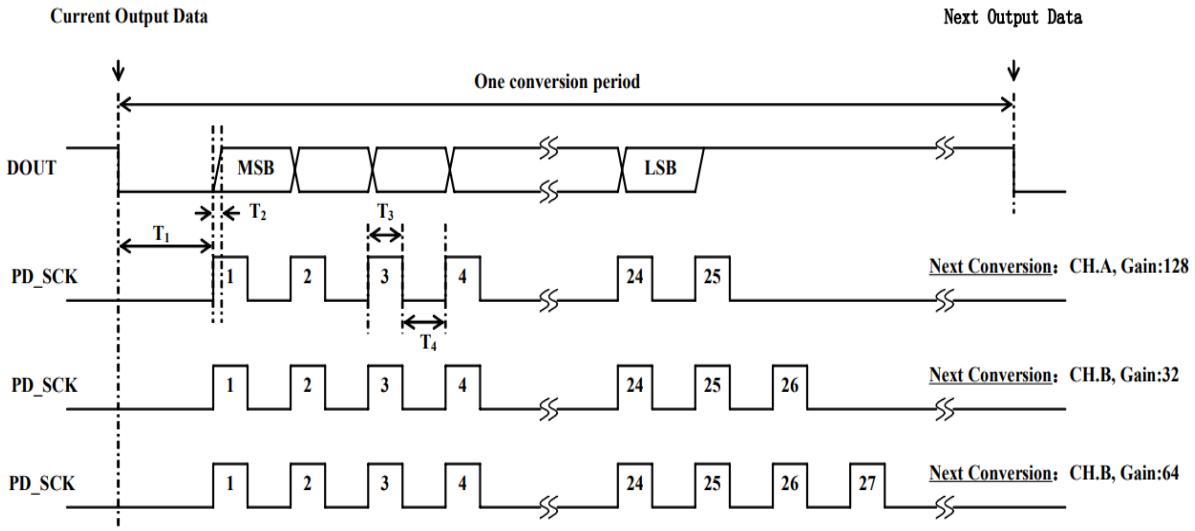


Figure 2.11 Depicts the timing diagram of the input data, output data, and gain factor

Symbol	Note	MIN	TYP	MAX	Unit
T_1	DOUT falling edge to PD_SCK rising edge	0.1			µs
T_2	PD_SCK rising edge to DOUT data ready			0.1	µs
T_3	PD_SCK high time	0.2	1	50	µs
T_4	PD_SCK low time	0.2	1		µs

Table 2.1 Presents the required time for the data transmission process

* Specifications:

+ Operating Voltage: 2.7 - 5V
+ Current Consumption: < 1.5 mA
+ Sampling Rate: 10 - 80 SPS (adjustable) (SPS: samples per second)
+ Resolution: 24-bit ADC
+ Voltage Resolution: 40mV
+ Dimensions: 38 x 21 x 10 mm

- Wiring diagram loadcell and HX711

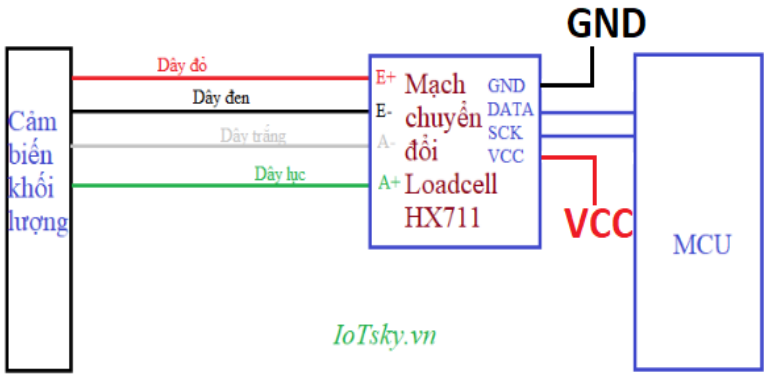


Figure 2.12 Wiring diagram HX711 module and loadcell

2.4 UART Serial Communication Protocol

2.4.1 Concept

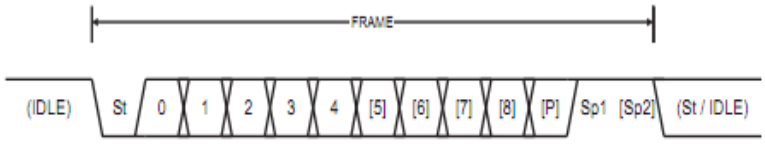
- UART - Universal Asynchronous Receiver Transmitter is an asynchronous serial communication interface. It is commonly an integrated circuit used for data transmission between a computer and peripheral devices.

2.4.2 Characteristics

+ Transfers 1 bit within a specified time unit at a predetermined data transfer rate (baud rate).
+ "0" level corresponds to 0 VDC voltage.
+ "1" level corresponds to voltage ranging from 3.3-5 VDC.

- Frame structure of the UART protocol:

Figure 19-4. Frame Formats



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

Figure 2.13 Data transmission frame

+ Start Bit: Always at a low level (logic "0") to indicate the beginning of the data frame.
+ Data Bits: 8 bits of data are transmitted. These bits carry the actual data to be transmitted.
+ Parity Bit: This is an optional bit used for error detection. It can be set to even parity or odd parity based on the chosen parity rule.
+ Stop Bit: Always at a high level (logic "1") to indicate the end of the data frame.
+ DLE (Data Link Escape): No data is transmitted on the communication line. The IDLE line state must be at a high level.

2.4.3 RS232 Standard (Recommended Standard 232):

a) Concept:

- The RS232 standard is one of the widely used techniques to connect peripheral devices to computers. It is an asynchronous serial communication standard that allows a maximum of two devices to be connected. The maximum allowable cable length to ensure reliable data reception is 15 meters, and the data transfer rate is up to 20 Kbit/s.

b) Characteristics:

Standard	Specification
Maximum Transmission Distance	15m (at baud rate = 9600)
Maximum Data Transfer Rate	20Kbps
Maximum Output Voltage	±25 VDC
Output Voltage with Load	± 5 VDC đến ±15 VDC
Load Impedance	3kΩ đến 7kΩ
Input Voltage	± 15 VDC
Input Sensitivity	± 3VDC
Input Impedance	3kΩ đến 7kΩ

Table 2.2 The RS232 standard communication

c) Some Concepts in UART (or TTL) Protocol:

- Baud Rate: The baud rate refers to the number of data packets (characters) transmitted per second. A data packet can consist of one or more bits. Commonly used baud rates in practice are 1200, 2400, 4800, 9600, 19200, and so on.
- Parity Bit: The parity bit is a error-checking bit used in the process of data transmission/reception, and it is a widely employed technique. Essentially, an additional bit is added to the transmitted data to detect or correct errors during transmission. Depending on the

total number of "1" bits in the transmitted data being even or odd, a "0" or "1" parity bit is added. The value of the parity bit can be selected as follows:

+ If even parity is chosen, the additional bit is "0" when the total number of "1" bits in the data packet is even.
+ If odd parity is chosen, the additional bit is "0" when the total number of "1" bits in the data packet is odd.

2.5 Overview of LM7805 and Power Supply

a) Power Supply Block

+ The power supply block ensures stable operation of the model once it is completed, and it is the most crucial block that needs to be accurately calculated during the model construction.
+ We observe that the power supply for the microcontroller is 5 VDC. Therefore, we choose a 5V/1A adapter to obtain a 5V power source. However, for this project, we use a power supply circuit to generate a stable 5V voltage similar to an adapter, as follows:

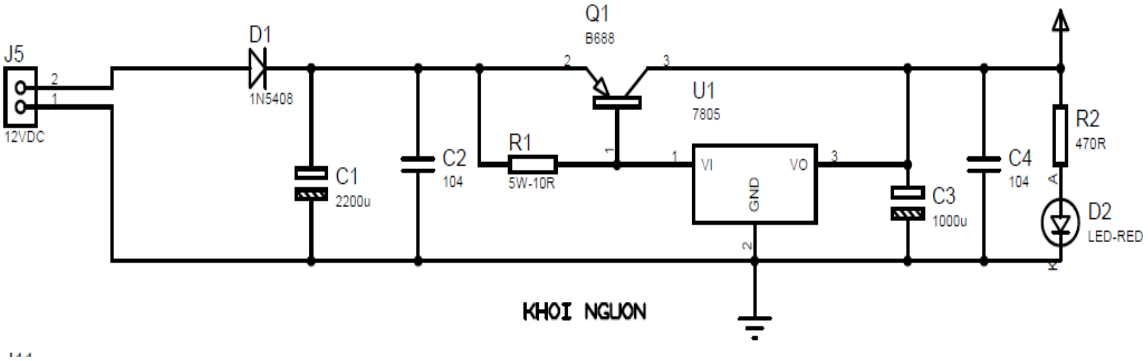


Figure 2.14 The power supply circuit provides power to the entire system

+ The input voltage of 12VDC will be regulated using the IC7805 to provide an output voltage of 5VDC.

b) IC 7805

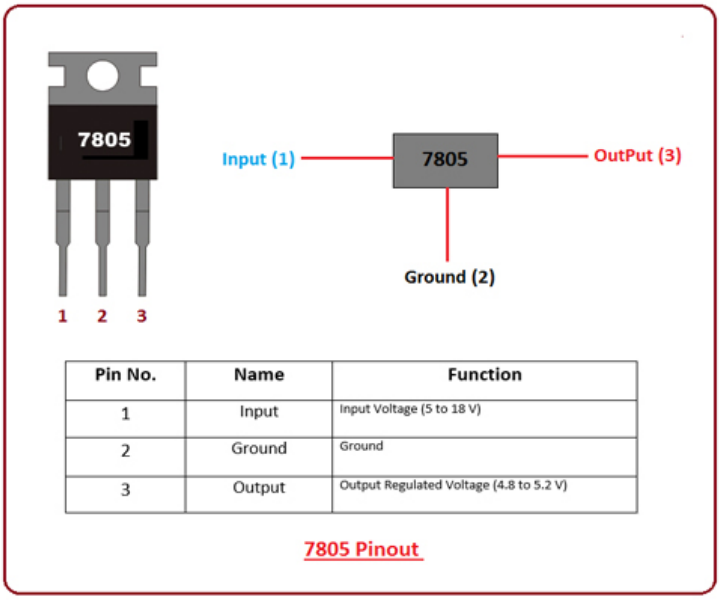


Figure 2.15 Pins diagram of IC7805

- Specifications:
  + Output Voltage: 5 V
  + Input Voltage: 7V - 18 VDC
  + Maximum Output Current: 1A
  + Output Current: 1A
  + Operating Temperature Range: 0ºC to 125ºC
  + Maximum Power Dissipation: 5W

- Function: LM7805, or IC 7805, is known as a voltage regulator IC that regulates the output voltage to +5V. The 7805 belongs to the LM78xx series of positive voltage regulators and is manufactured in a TO-220 package. This IC is widely used in various electronic circuits and commercial electronic devices.

2.6 Introduction to Webcam and QR Code

2.6.1 Webcam

a) Definition:

- A webcam is a device that combines the words "web" and "camera" and is often abbreviated as "WC." It is used to assist laptop users in capturing live video, making it easier to view images and communicate with distant relatives, partners, colleagues, friends, etc., through the internet.

- Nowadays, webcams are an essential part of technology devices in general and have diverse applications in daily life. With impressive video chat features, they provide unexpectedly enjoyable experiences.

* Uses of Webcam:
  + Home surveillance and video call
  + Face recognition
  + Gaming support
  + Barcode scanning
  + Optical character recognition

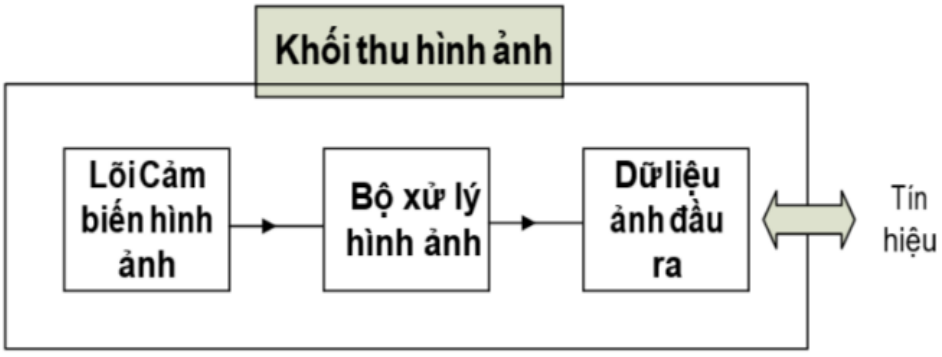


Figure 2.16 Camera block diagram

- Webcams can be used for recording high-definition videos as well as capturing still images. They are relatively easy to use for beginners, but there are also many advanced solutions

available to meet the demands of advanced users. There are numerous user demos showcasing the functionality of camera modules, such as time-lapse photography, slow-motion videos, and many other applications.

+ Connect the webcam using a 15cm USB cable to the laptop's USB port.
+ With this setup, users can use commands to capture images and videos from the camera.

- From the advantages and benefits, we decided to use the webcam from the discarded female student's laptop and reuse it to use for this project similar to the webcam sold in the market at a cheap price. Good differential analysis, fairly cheap price, cost-effective, lower product cost.

Figure 2.17 Old Dell laptop with camera for project

- Technical specifications:
  + HD720p resolution (1280 x 720) pixel
  + Operating voltage: 5V
- Connection diagram:

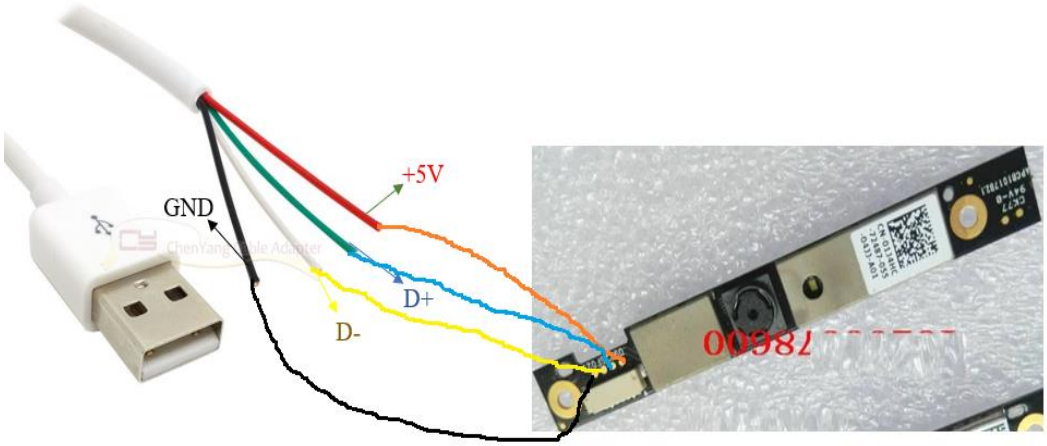


Figure 2.18 USB and Webcam connection diagram

* Applications:

+ We can use the webcam and specialized software to set up a QR code object detection system. This system operates by using images from the webcam combined with image processing algorithms to detect QR codes. It can then trigger specific commands or actions based on the detected QR codes.

+ The webcam is also particularly useful for filmmakers, allowing them to capture challenging angles or unique shots that may not be achievable with larger cameras. Additionally, you can utilize the webcam to record time-lapse videos (by stitching multiple images together), a popular technique employed by many users worldwide.

2.6.2 Introduction to QR Codes

- QR Code (Quick Response code), also known as a matrix barcode, is a two-dimensional barcode that can be read by specialized barcode scanners or smartphones equipped with barcode scanning applications such as the camera or popular messaging apps like Zalo.
- A QR Code consists of black dots and squares arranged within a larger square. The size of the QR Code may vary depending on the specific product or use case. QR Codes are designed to replace traditional linear barcodes. They are faster to read, save time, and require less space compared to traditional barcode formats.
- QR Codes can contain various types of information, such as website addresses (URLs), email addresses, SMS messages, event schedules, text content, or geographic location information. Depending on the specific QR Code, it can hold different types of content, and each content corresponds to a unique QR Code. In this project, we will utilize the QR Code functionality to store the IDs of the products.
- By scanning a QR Code, users can quickly access the information embedded within it, making QR Codes a convenient and versatile tool for sharing and retrieving data in various applications.

Figure 2.19 QR code illustration

+ When scanning a QR Code image, you will receive a corresponding sequence of numbers that represents the ID of the product or any other value that was encoded into the QR Code. The specific values encoded in the QR Code depend on the creator or the initial setup of the QR Code. However, in this project, we are referring to generating a sequence of ID numbers.

+ Creating a QR Code nowadays is easy and convenient. You can simply search on Google using the keyword "QR code generator," and you will find various online tools that allow you

to generate your own QR Codes. Furthermore, many of these QR code generator tools are available for free, making it even more accessible to create QR Codes for your specific needs.

2.7 Overview of Image Processing

- Image processing is a scientific and technological field. It is a relatively new discipline compared to many other scientific fields, but it is rapidly developing, stimulating research centers, applications, especially dedicated computers for it.

- Image processing is a technique applied to enhance and process images captured by devices such as cameras, webcams, etc. Therefore, image processing has been applied and developed in many important fields such as:

  - In the field of security: Face recognition, fingerprint recognition, iris patterns, human images, and other devices.
  - In the military field: Processing and identifying military equipment, serving reconnaissance, monitoring important targets.
  - In the entertainment industry: Serving the filmmaking process and creating electronic games, graphic processing.
  - In the medical field: Processing X-ray images, MRI, biomedical image processing, etc.
  - In the field of AI: Image recognition and processing for robots, human-robot interaction, graphic processing.

a) Images in computers

- Color images: The RGB color system stands for red, green, blue, which are the three primary colors of light when separated through a lens. By mixing these three colors in certain proportions, various other colors can be created

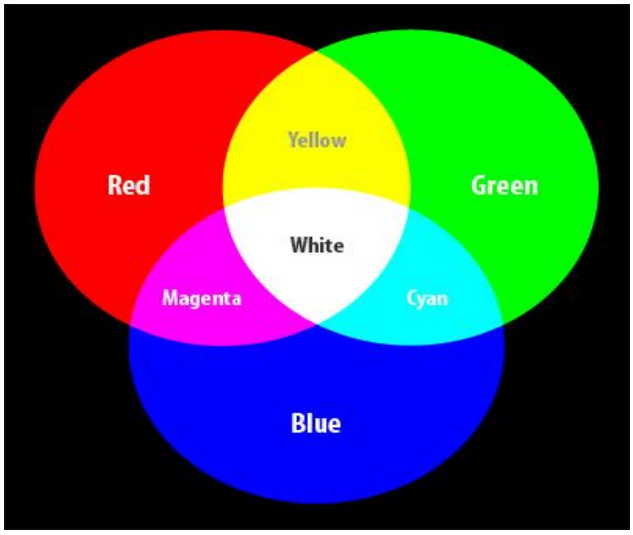


Figure 2.20 Description of the RGB color space

+ Adding red to green creates yellow; adding yellow to blue creates white. For each set of three integer values r, g, b in the range [0, 255], a different color is produced. With 256 possible choices for red, 256 for green, and 256 for blue, the total number of colors that can be produced in the RGB color system is: 256 * 256 * 256 = 16,777,216 colors.

+ To facilitate storage and processing, it is not possible to store them in a matrix like that, so each value in each pixel will be separated into a separate array, forming a 3-dimensional array (tensor).

$$\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,800} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,800} \\ \cdots & \cdots & \cdots & \cdots \\ r_{600,1} & r_{600,2} & \cdots & r_{600,800} \end{bmatrix}, \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,800} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,800} \\ \cdots & \cdots & \cdots & \cdots \\ g_{600,1} & g_{600,2} & \cdots & g_{600,800} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,800} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,800} \\ \cdots & \cdots & \cdots & \cdots \\ b_{600,1} & b_{600,2} & \cdots & b_{600,800} \end{bmatrix}$$

Figure 2.21 Photo 600 pixels wide and 800 pixels high

* Represented as a matrix of 600*800

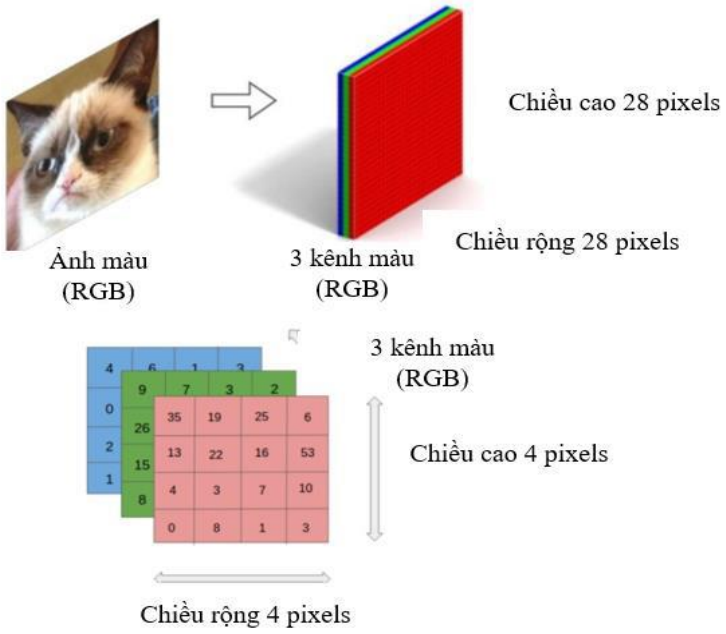


Figure 2.22 Color image tensor order 3

b) Gray image

+ Similar to a color image, a gray image also has a size of 800 pixels * 600 pixels, which can be represented as a matrix of size 600 * 800 (meaning the number of rows multiplied by the number of columns).

+ However, each pixel in a gray image only needs to be represented by an integer value in the range [0, 255] instead of (r, g, b) as in a color image. Therefore, when representing a gray image in a computer, a single matrix is sufficient.

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,800} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,800} \\ \cdots & \cdots & \cdots & \cdots \\ w_{600,1} & w_{600,2} & \cdots & w_{600,800} \end{bmatrix}$$

The value 0 represents black, 255 represents white, and the closer the pixel value is to 0, the darker it is, and the closer it is to 255, the brighter it is.

c) Color space conversion for an image

- Each pixel in a color image is represented by 3 values (r, g, b), while in a gray image, only one value, x, is needed for representation.

When converting from a color image to a gray image, we can use the formula:

x = r * 0.299 + g * 0.587 + b * 0.114

+ However, when converting back, because we only know the value of x and need to find r, g, b, it will not be accurate.

2.8 Analysis of functional blocks in the system:

a) Loadcell Sensor:

+ This sensor detects weight and produces corresponding voltage signals.

b) HX711 Module:

+ It converts the analog value (voltage) from the loadcell into a digital value and amplifies it before transmitting it to the microcontroller.

c) PIC16F877A Microcontroller:

+ It acts as the central processing unit, responsible for reading data from the HX711, processing the results, and determining the weight.

+ It transmits data to the computer through Serial communication and a USB converter.

d) Computer:

+ It plays a crucial role, providing software tools for programming the PIC16F877A, using Visual Studio to program the system's interface, and communicating with a webcam for QR code processing.

CHAPTER 3 HARDWARE DESIGN AND PROGRAMMING IMPLEMENTATION FOR MICROCONTROLLER

3.1 Hardware Design for the Electronic Scale Model

+ 3D Digital Scale Model

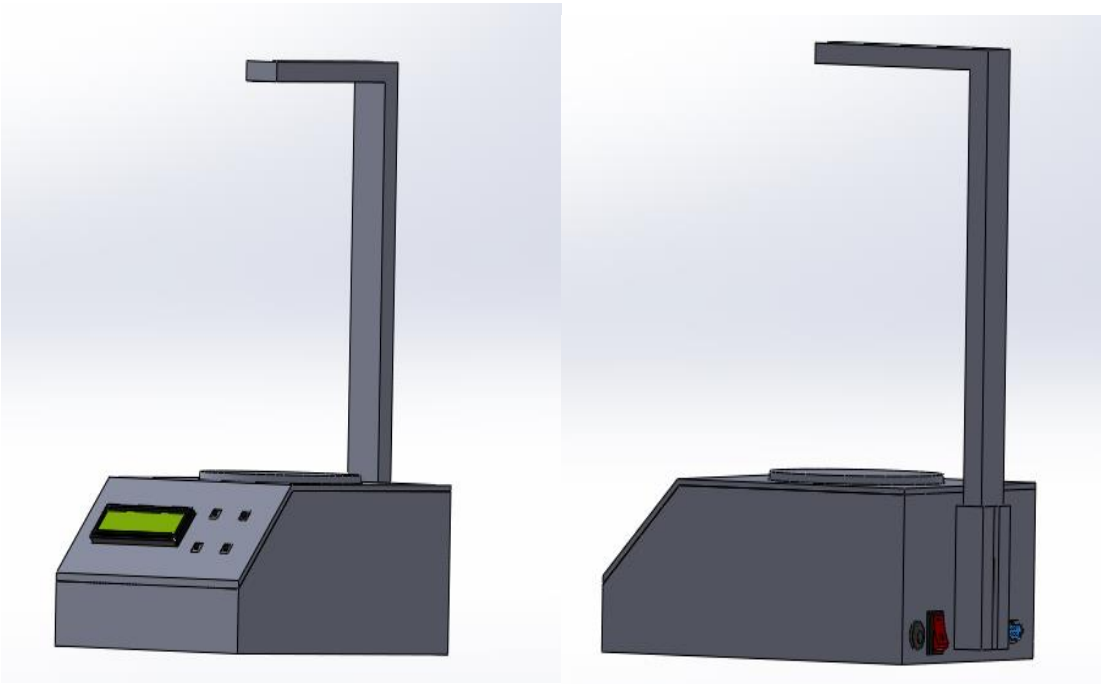


Figure 3.1 Overall shape of the model

+ Dimensions of each component:

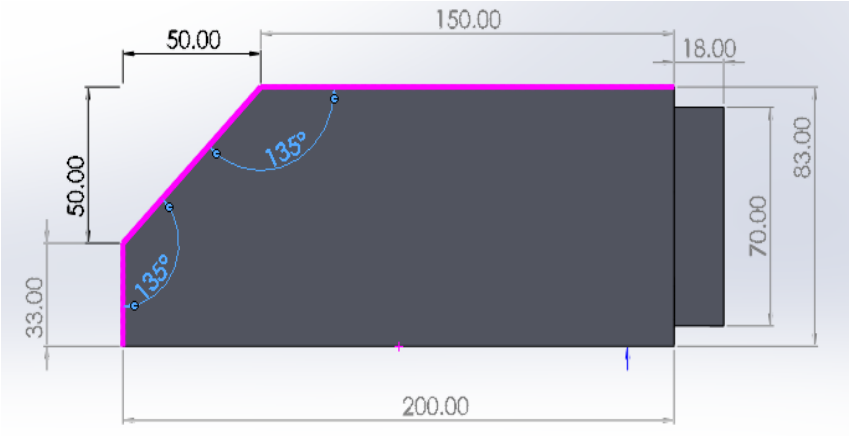


Figure 3.2 Side view of the model

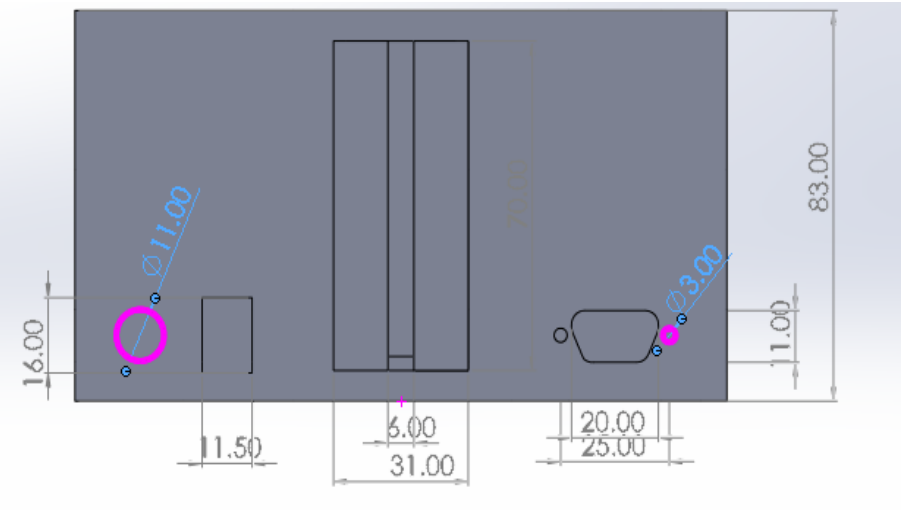


Figure 3.3 The back of the model

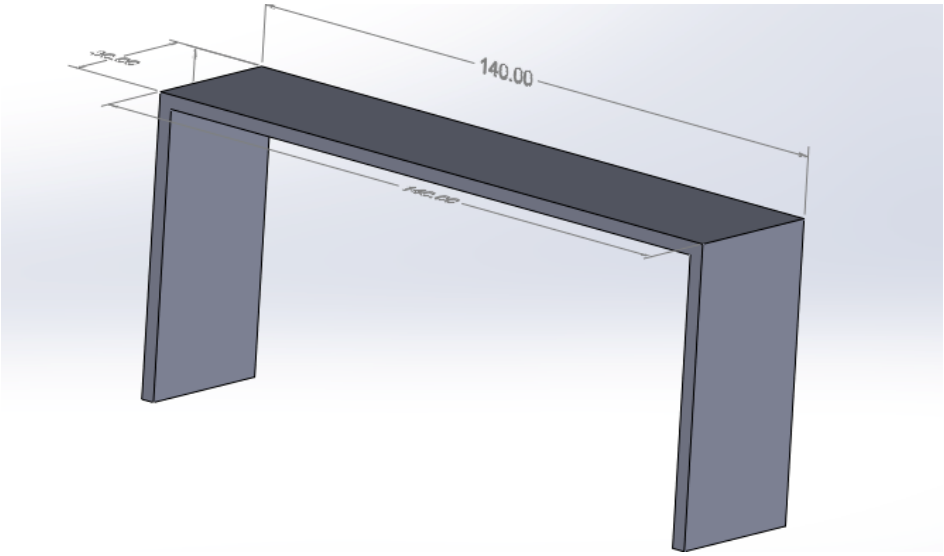


Figure 3.4 The sensor bracket

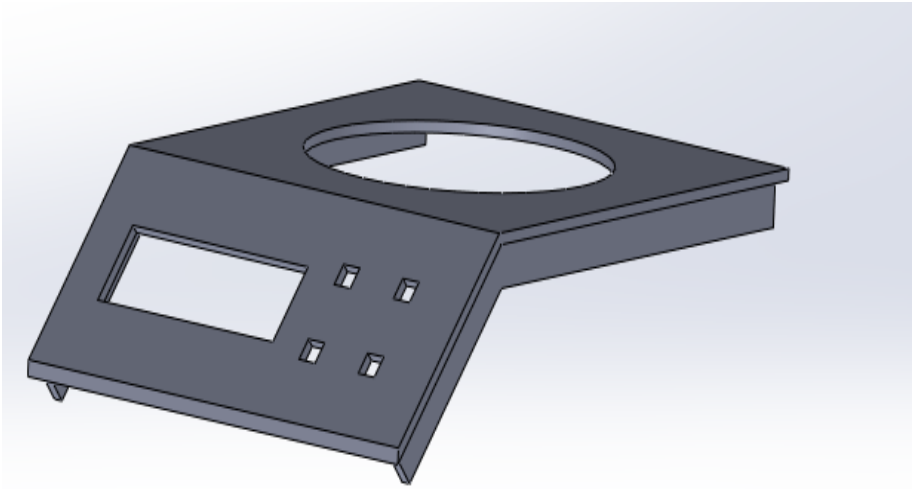


Figure 3.5 The cover of the model

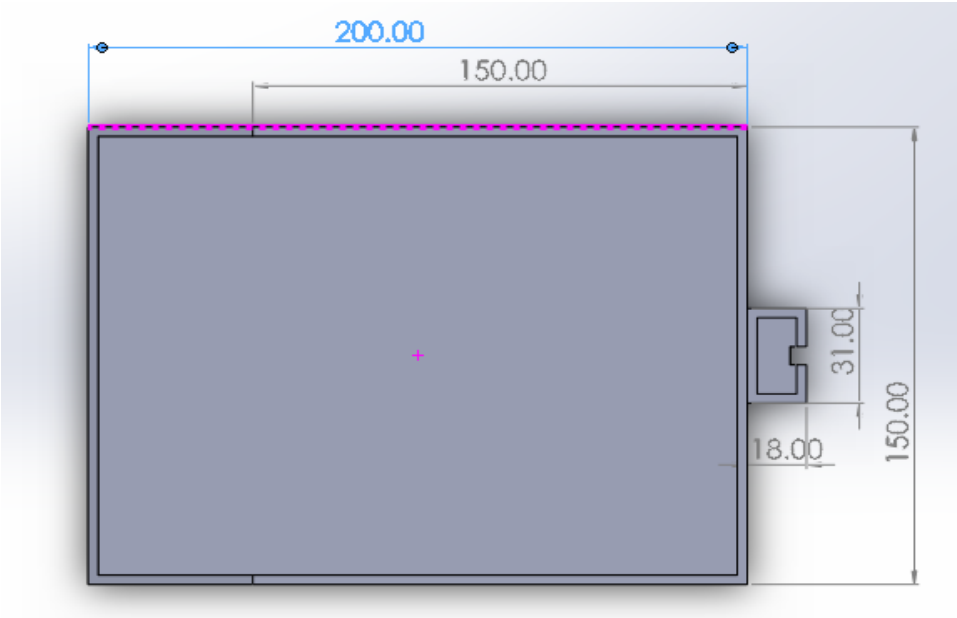


Figure 3.6 The bottom surface of the model

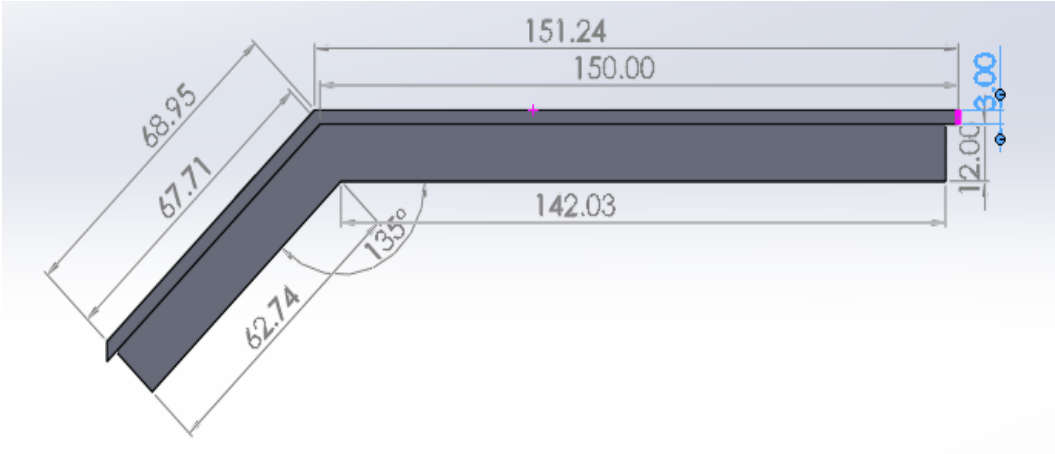


Figure 3.7 The side projection of the cover

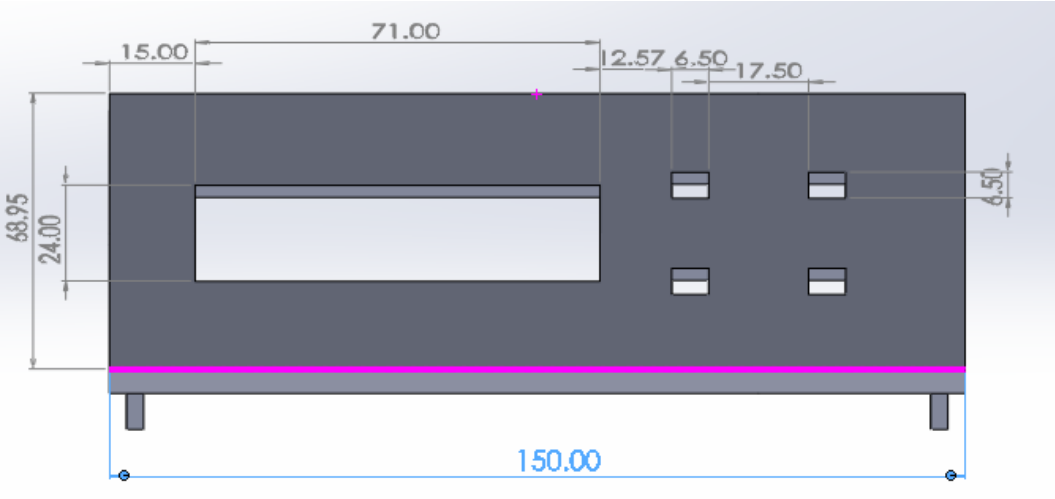


Figure 3.8 The front vertical projection of the cover

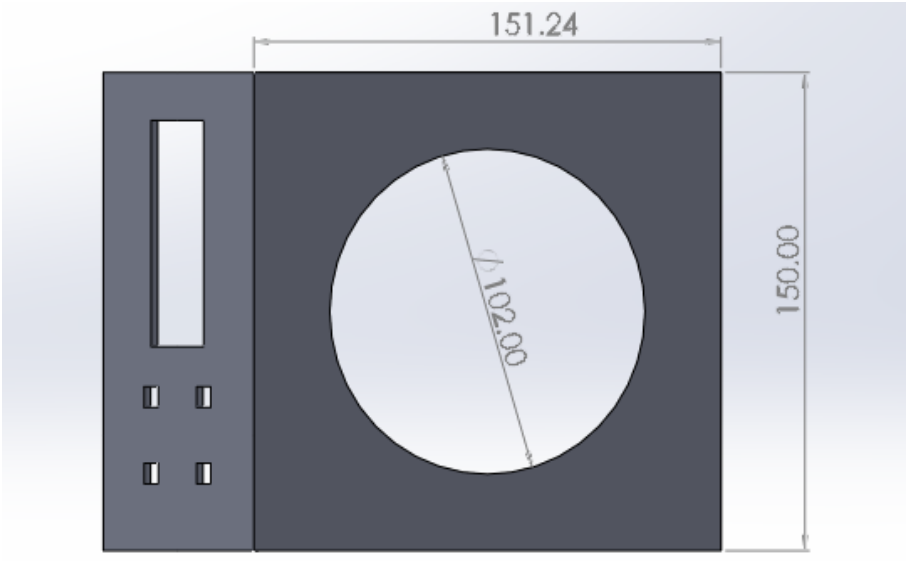
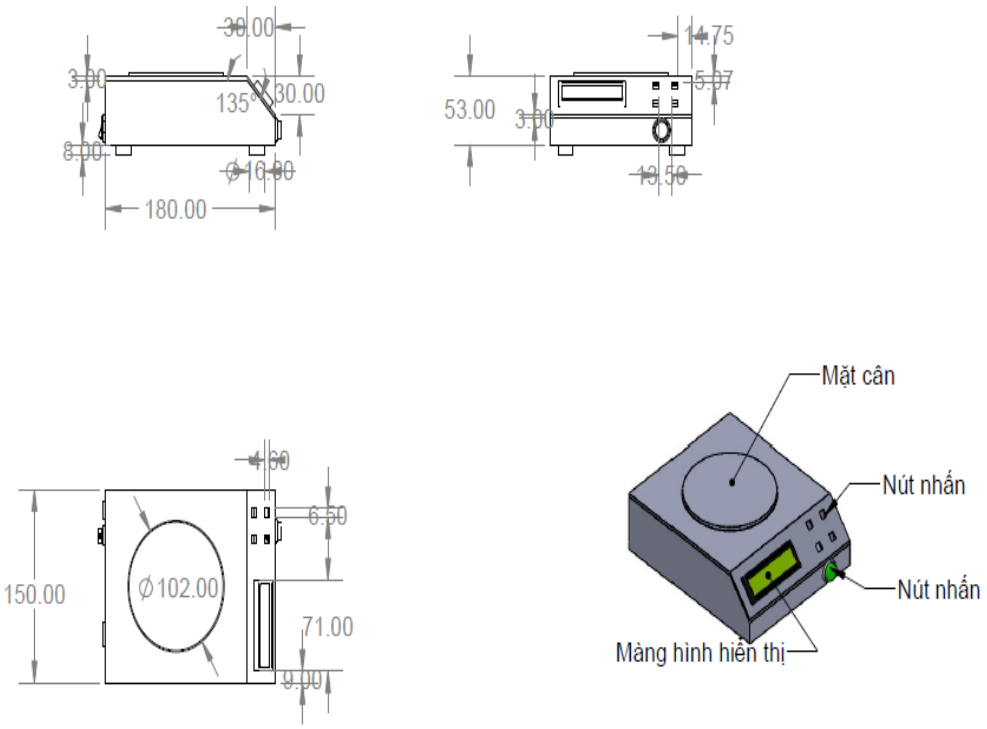


Figure 3.9 The weight scale of the cover



vỏ cân loadcell		Vật liệu	Tỉ lệ	Đơn vị
		Nhựa PVC	1:2	mm
Người vẽ	Phan Hữu Thắng	Trường Đại học Bách khoa Đại học Đà Nẵng		
kiểm tra				

Figure 3.10 The overall schematic diagram of the scale

3.2 Building the electrical circuit

3.2.1 General introduction to the electrical circuit

+ Block diagram

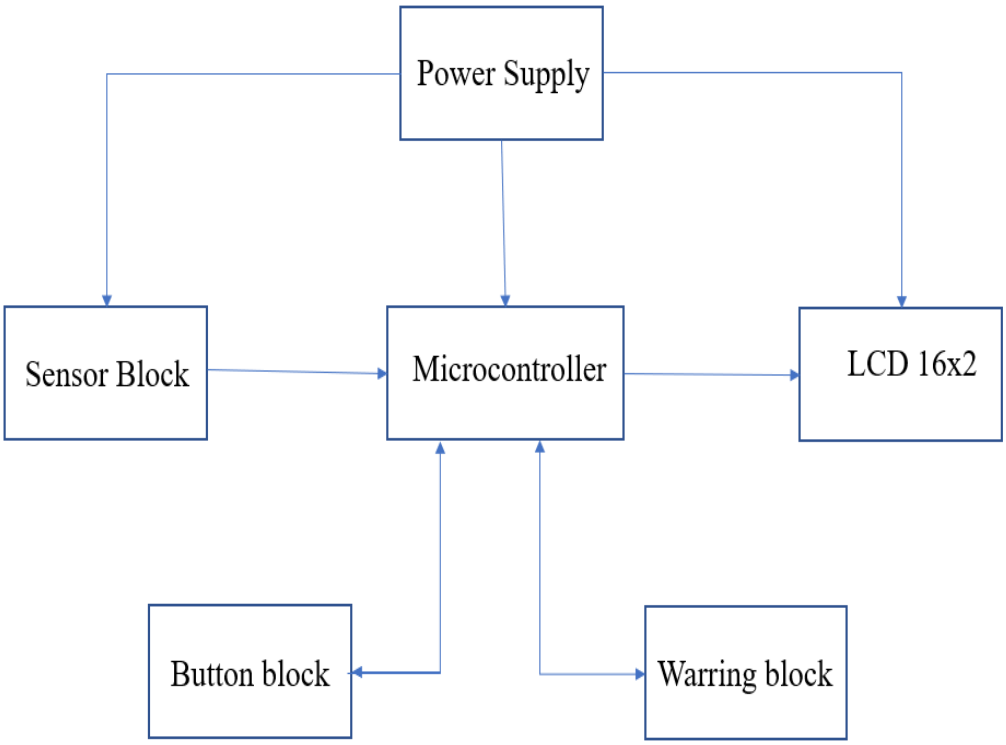


Figure 3.11 System block diagram

- The power block is used to supply voltage and current to the entire electrical circuit.
- The sensor block is used to read the specific weight value from the sensor and send it to the microcontroller for processing.
- The specific microcontroller used is PIC16F877A, which is responsible for controlling and monitoring the devices that can communicate with a computer through the COM port.
- The LCD display block is responsible for displaying the operational interface on a 16x2 LCD screen.
- The button block transfers input signals from the buttons to the microcontroller for processing.
- The warring block receives signals from the microcontroller and power source to output alarm signals.

2) Circuit design

a) Power circuit

+ The power supply for the power circuit is obtained from a transformer with the form U = $12\sqrt{2}\sin\omega t$. This voltage is then passed through diode D1, which is responsible for preventing reverse current flow in the circuit. The voltage then goes through capacitors C1 and C2, which are used to smooth out any unstable voltage fluctuations or ripples. After that, this voltage will

be converted into power to convert electrical energy into heat energy. This process helps reduce the magnitude of the current passing through the resistor and B688 transistor, which amplifies the current to meet the necessary requirements of the electrical circuit.

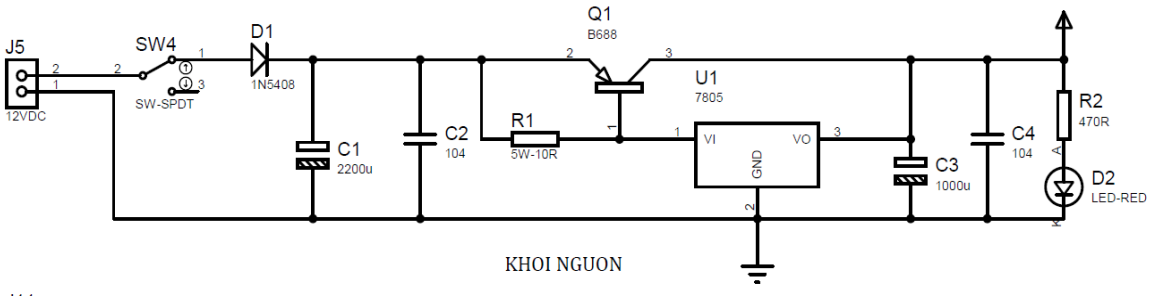


Figure 3.12 Power circuit diagram

+ After that, the voltage passes through IC7805 to obtain a regulated output voltage of 5V, which is used to supply the entire system. To ensure a stable output voltage, capacitors C3 and C4 are added at the back. Their purpose is to filter out any remaining ripples or disturbances in the voltage, thus achieving a consistently stable output voltage.

b) Sensor block

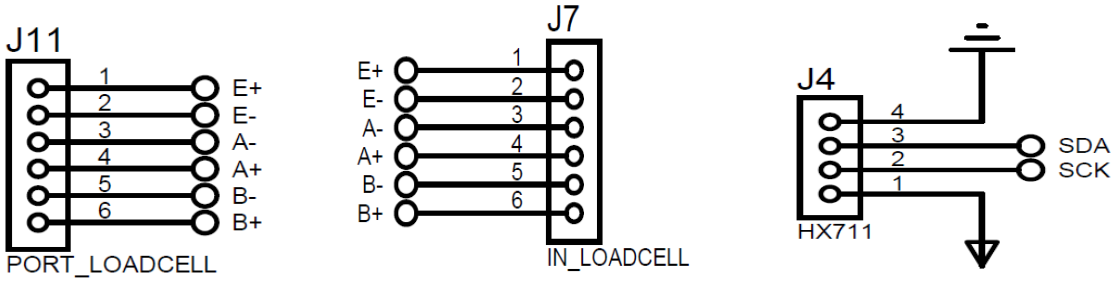


Figure 3.13 Pins diagram of HX711 and loadcell

+ The pinout diagram of the HX711 module shows that the two signal wires from the load cell (+Sig and -Sig) are connected to the A+ and A- pins of the HX711 module. The remaining two pins (E+ and E-) are used for power and ground connections. Additionally, the B+ and B- pins can be used depending on the amplification factor required.

+ The function of the HX711 module is to amplify the signal from the load cell and convert it from analog to digital format.

c) Display block

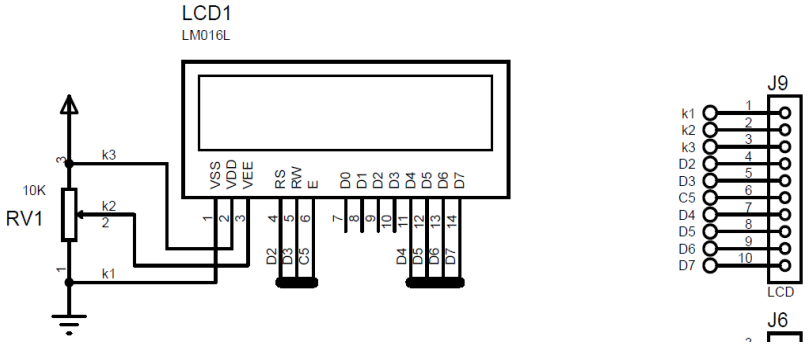


Figure 3.14 Pins diagram of the LCD

d) Microcontroller block

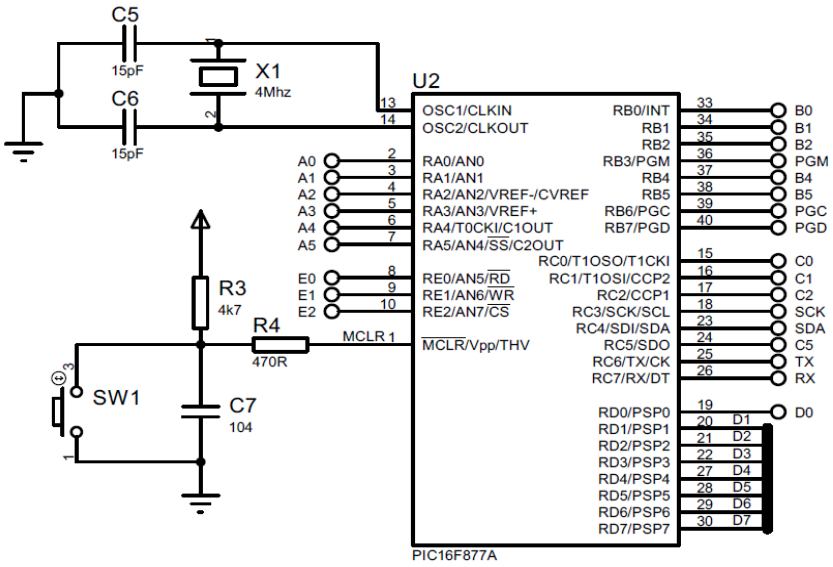


Figure 3.15 Microcontroller connection pin diagram

- We utilize several pins of the microcontroller to perform various functions, as follows:
  - For code programming, we use the PGM, PGC, PGD, and MCLR pins.
  - To receive signals from the HX711 module, we use the SCK and SDA pins.
  - The crystal oscillator block is connected to pins 13 and 14, consisting of a 4MHz crystal and capacitors C5 and C6 to generate a 1MHz frequency. Each instruction cycle of the microcontroller consumes 1µs.
  - The B0 - B3 pins serve as input pins to transfer signals from buttons to the microcontroller.
  - For UART communication, the TX and RX pins are used to communicate with a computer via the COM port.
  - The reset button is connected to the MCLR pin (pin 1) to reset the input signals for the microcontroller.
  - Port D pins, from D2 to D7, and pin C5 are connected to the LCD.
  - D1 pin is connected to the speaker signal block, while D0 and C2 pins are connected to two dot LEDs.
  - Some remaining pins are reserved for future use or as backup.

e) Warring block

Figure 3.16 The LED display and the buzzer

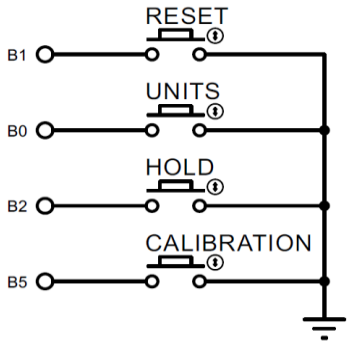


Figure 3.17 Function button

f) Com port

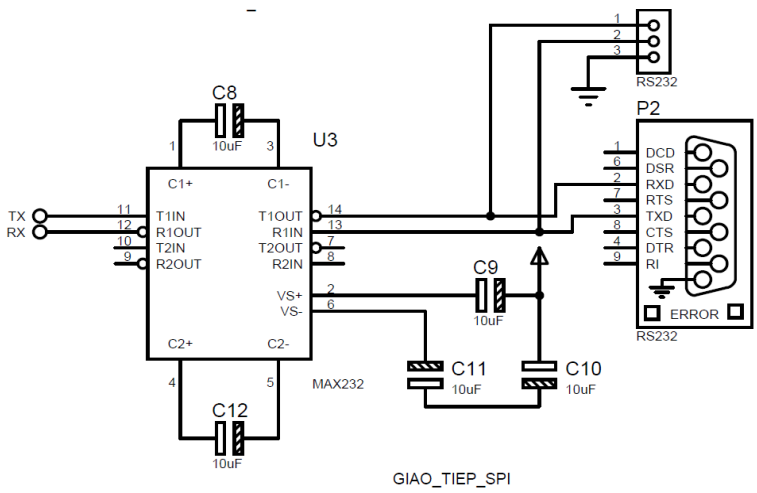


Figure 3.18 To communicate with a computer

g) The schematic drawing of the system

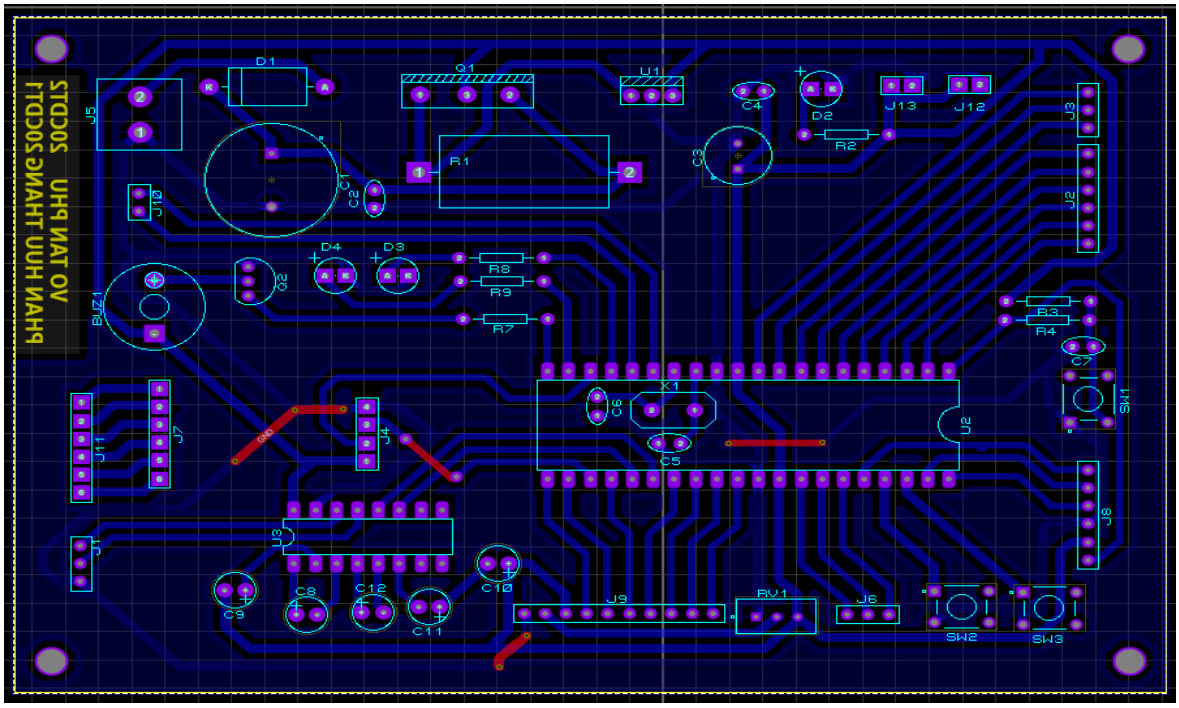


Figure 3.19 PCB layout drawing

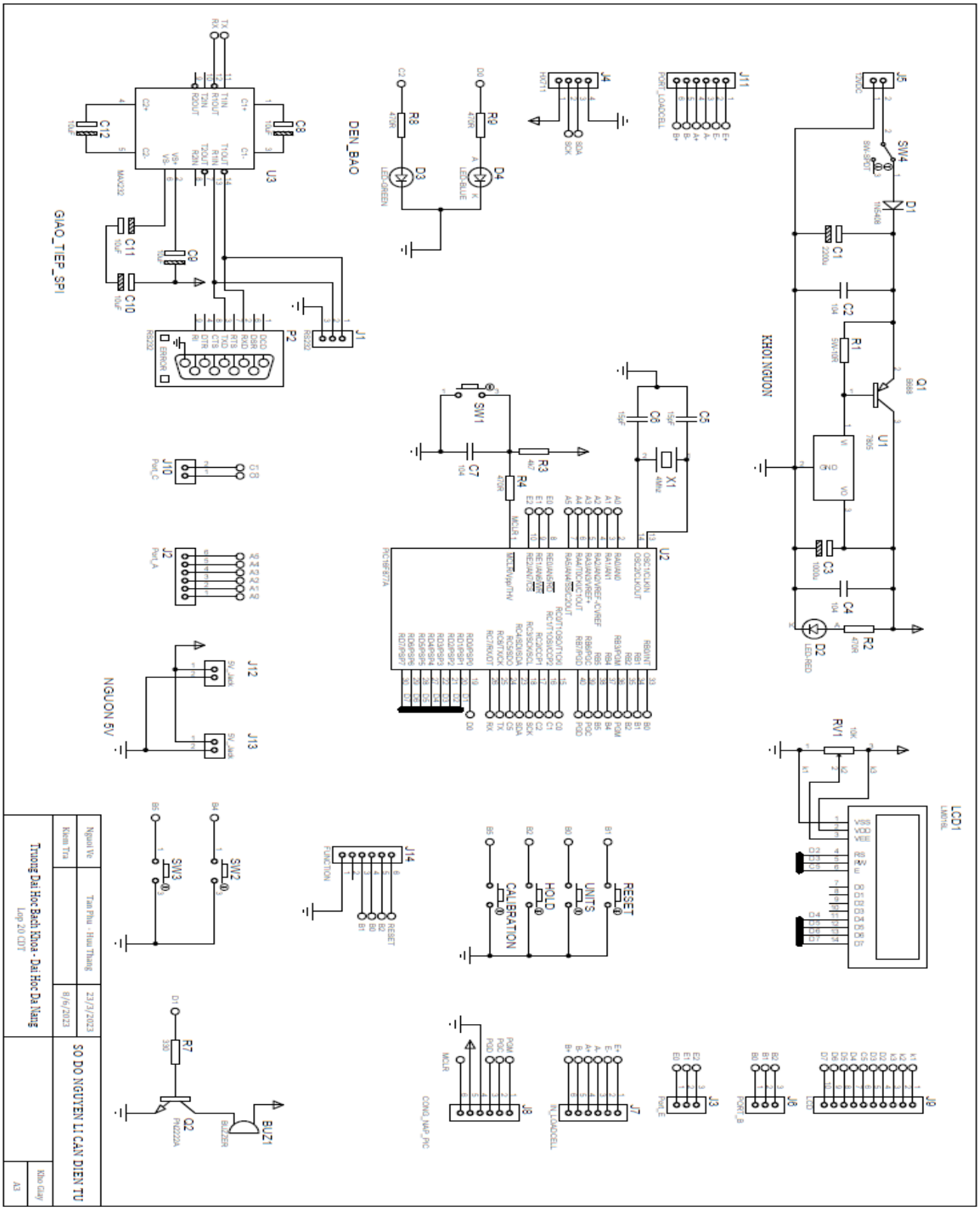


Figure 3.20 Diagram of principle

3.3 Programming for the system

3.3.1 Microcontroller programming using CCS

a) Algorithm flowchart

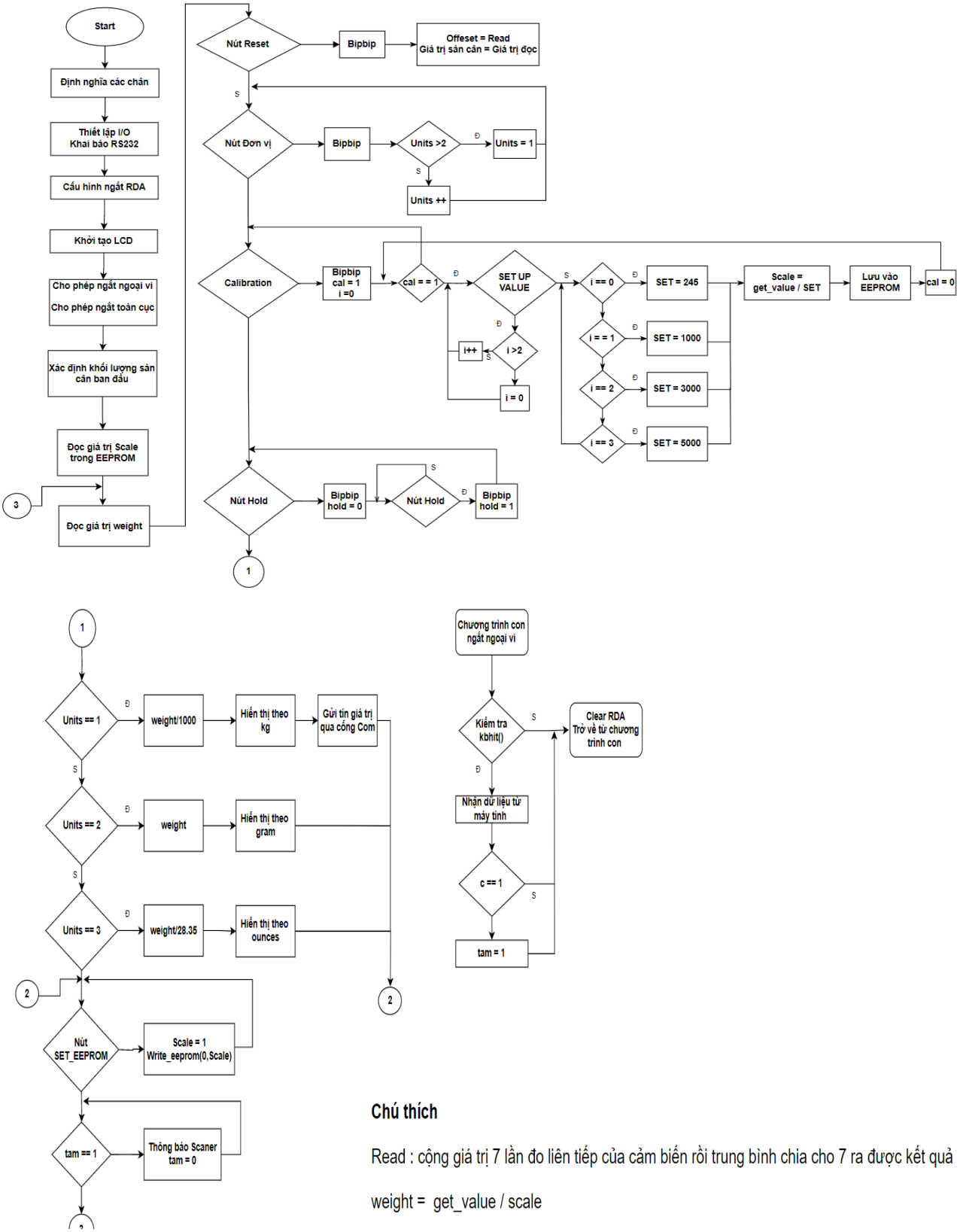


Figure 3.21 Program flowchart

b) Working principle

+ Initially, when there is no object on the weighing scale, the display will show a value of 0. When an object is placed on the scale, the load cell will bear the applied force, causing the strain gauge resistance to change. The sensor will send the voltage signal through the HX711 for amplification and analog-to-digital conversion, and then send it to the microcontroller for processing and display on the LCD screen.
+ In addition, the system has some additional features. When certain buttons are pressed, the system will emit sound signals through a speaker. For example, the RESET button is used to reset the displayed sensor value to 0, the HOLD button is used to freeze the displayed value on the LCD, the UNITS button is used to switch the measurement units of the scale (kg, g, oz), and the Calibration button is used to calibrate the sensor to accurately measure values.
In the calibration process, first, press the RESET button to reset the scale to 0.
+ Then, press the Calibration button and choose a predefined standard weight value, such as 245 g, 1 kg, 3 kg, or 5 kg. Simply place the corresponding weight on the scale. Press the RESET button (the value increase button in this function) to select the value and place the standard weight on the scale. Then, press the UNITS button (the value selection button in this function) to select the corresponding value for scaling. The microcontroller will calculate and store the calibration value. Turn off and turn on the system again to update the new scaling value. Verify the results against the standard weight value to complete the calibration process.
c) Code CCS

```
#include <main.h>
#use delay(crystal = 4MHz)
#FUSES HS,NOPROTECT,NOWDT,NOBROWNOUT,NOPUT,NOLVP
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, bits=8, parity=N)
#include "hx711.c"
#define RESET       PIN_B1              // Reset
#define DON_VI      PIN_B0               // Nut don vi
#define LCD_ENABLE_PIN  PIN_C5
#define LCD_RS_PIN      PIN_D2
#define LCD_RW_PIN      PIN_D3
#define LCD_DATA4       PIN_D4
#define LCD_DATA5       PIN_D5
#define LCD_DATA6       PIN_D6
#define LCD_DATA7       PIN_D7
#include <lcd.c>                    // Khai bao thu vien Lcd
float Scale;
int8 tam;
float SET_UP[4] = {245,1000,3000,5000};    // Cac gia tri hieu chuan
#INT_RDA                // Ngat UART ngoai vi - khi tinh hieu
void RDA_isr()
```

```
{
    if(kbhit())
    {
      char c = getch();  // doc 1 chuoi ki tu
      if(c=='1')
      {
         tam = 1;
      }
    }
    clear_interrupt(INT_RDA);                  // xoa bo co ngat
}
void bipbip()
{
    output_high(PIN_D1);
    delay_ms(100);
    output_low(PIN_D1);
}
void WRITE_FLOAT_EEPROM(long int n, float data)         // ghi gia tri vao trong bo nho
EEPROM cua VDK ( dia chi, &gia tri)
{
  for (int i = 0; i < 4; i++)
  {
     write_eeprom(i + n, *((int8*)&data + i) ) ;      //  (int8*)&data try cap vao dia chi cua data
+ try cap vao byte thu i
  }
}
float READ_FLOAT_EEPROM(long int n)                  // Doc gia tri trong bo nho cua VDK
{
  float data;
  for (int i = 0; i < 4; i++)
  {
    *((int8*)&data + i) = read_eeprom(i + n);
  }
  return(data);
}
// chuc nang chuan CB
void calibration()
{
 int i = 0, cal=1;
```

```
  lcd_gotoxy(2, 1);
  printf(lcd_putc, "Calibracion");
  lcd_gotoxy(4, 2);
  printf(lcd_putc, "Loading...");
  delay_ms(500);
  tare(7);
  lcd_putc("\f");
  // Start qua trinh hieu chuan
  while(cal == 1)
  {
    lcd_gotoxy(1, 1);
    printf(lcd_putc, "SET UP KL:");
    lcd_gotoxy(1, 2);
    printf(lcd_putc, "     %4.0f g        ",SET_UP[i]);

    // Ghi gia tri trong luong da biet bang nut RESET
    if(input(RESET) == 0)
    {
      while(input(RESET) == 0)
      i =(i>2) ? 0:i+1; //tuong tu nhu cau lenh if-else ( if i > 2 gan i = 0 else gan i = i+1 ) cau
lenh so sanh
    }
    //Select trong luong da biet bang nut DON_VI
    if(input(DON_VI) == 0)
    {
      while(input(DON_VI) == 0);
      lcd_putc("\f");
      lcd_gotoxy(1, 1);
      printf(lcd_putc, "Set trong luong");
      lcd_gotoxy(1, 2);
      printf(lcd_putc, "Loading....");
      delay_ms(1000);
      Scale = (float)(get_value(7)/ SET_UP[i]);       // Gia tri doc chia cho trong luong da biet
      WRITE_FLOAT_EEPROM(0, Scale);           // Save in EEPROM
      delay_ms(100);
      cal = 0;                                        // Gan cal = 0 de ket thuc vong lap
      lcd_putc("\f");                                 // xoa man hinh
    }
  }
```

```
}
void Hold_value()
{
  int hold = 0;
  bipbip();
  do
  {
     if(input(PIN_B2)==0)
     {
       while(input(PIN_B2)==0);
       bipbip();
       hold = 1;
     }
  }
  while(hold == 0);
}
void main()
{
    set_tris_c(0x80);
    set_tris_b(0x37);
    set_tris_d(0);
    port_b_pullups(1);
    output_low(PIN_D1);
    output_low(PIN_D0);
    output_low(PIN_C2);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    float weight = 0, factor = 1;
    int Units = 1;
    lcd_init();
    delay_ms(100);
    lcd_putc("\f");
    Scale = READ_FLOAT_EEPROM(0);              // Doc gia tri Scale trong EEPROM
    lcd_gotoxy(1, 1);
    printf(lcd_putc, "Loading...");
    set_scale(Scale);
    tare(7);                                   // Gan gia tri san can ban dau Offset
    delay_ms(300);
    lcd_putc("\f");
```

```
while(true)
{
    weight = get_value(7);
    lcd_gotoxy(3, 1);
    printf(lcd_putc, "CAN DIEN TU");
    switch (Units)
    {
        case 1:
            factor = 1000.0;
            lcd_gotoxy(1, 2);
            printf(lcd_putc, "KL: %4.3f Kg       ",weight/factor);

            printf("%4.3f\r",weight/factor);  // gui du lieu can nang qua cong Com
            delay_ms(1000);
            break;
        case 2:
            factor = 1.0;
            lcd_gotoxy(1, 2);
            printf(lcd_putc, "KL: %4.1f g       ",weight/factor);
            break;
        case 3:
            factor = 28.35;
            lcd_gotoxy(1, 2);
            printf(lcd_putc, "KL: %4.2f oz       ",weight/factor);
            break;
    }
    if(input(RESET)== 0)
    {
        while(input(RESET)== 0);
        bipbip();
        tare(7);
    }
    else if(input(DON_VI) == 0)
    {
        while(input(DON_VI) == 0);
        bipbip();
        Units = (Units>2)? 1:Units+1;
    }
    else if(input(PIN_B4)==0)     // Che do chuan cam bien cau CAN
```

```
        {
            while(input(PIN_B4)==0);
            bipbip();
            lcd_putc("\f");
            calibration();
        }
        else if(tam ==1)
        {
            bipbip();
            tam =0;
        }
        else if(input(PIN_B2)==0)
        {
          while(input(PIN_B2)==0);
          Hold_value();
        }
        else if(input(PIN_B5)==0)
        {
          while(input(PIN_B5)==0);
          bipbip();
          Scale = 1;
          WRITE_FLOAT_EEPROM(0, Scale);
        }
    }
}
*File hx711.c
#define DOUT       PIN_C4
#define SCK        PIN_C3
unsigned int32 OFFSET = 0;              // used for tare weight
float  tyle;              // used to return weight in grams, kg, ounces, whatever
void set_scale(float scale1)
{
    tyle = scale1;
}
unsigned int32 hx711_read()
{
  unsigned int32 Count = 0;
  unsigned int8 i;
  output_high(DOUT);             // Dua chan DOUT lên muc "1"
```

```
  output_low(SCK);            // Dua chan PD_SCK xuong muc 0"
  Count = 0;
  while(input(DOUT));
  for (i = 0 ; i < 24 ; i++)
  {
    output_high(SCK);
    Count = Count << 1;
    output_low(SCK);
    if(input(DOUT)) Count++;
  }
  output_high(SCK);           // Tao xung clock thu 25
  Count = Count^0x800000;           // Thuc hien phep toan logic OR
  output_low(SCK);
  return(Count);
}
int32 get_average(int times)
{
    unsigned int32 sum = 0;
    for (int i=0;i<times;i++)
    {
        sum = sum + hx711_read();
    }
    return(sum / times);
}
void tare(int times)
{
    OFFSET = get_average(times);   // reset nhan nut san can = gia tri cb doc
}
float get_value(int times)
{
    float value;
    unsigned int32 read = get_average(times);
    if(OFFSET >= read)
    {
        value =  (OFFSET - read)*(-1);
    }
    else
        value = read - OFFSET;
    return(value/tyle);
```

}

CHAPTER 4 DESIGN THE SYSTEM'S SALES MANAGEMENT INTERFACE AND DATABASE

4.1 Introduction to Visual Studio 2022 and SQL Server 2022 Software

4.1.1 Visual Studio 2022 Software

- Visual Studio 2022 is a relatively new version of the Visual Studio software provided by Microsoft. This software can help in programming and creating user interfaces (software applications) in a relatively easy and convenient manner. In this project, the user interface for the electronic scale will be developed using Visual Studio software using the C# programming language.

Figure 4.1 Logo Visual Studio

4.1.2 Microsoft SQL Server 2022 Software

- Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product that primarily functions to store and retrieve data as requested by other software applications. It can run on the same computer or on a different computer over a network (including the Internet).

Figure 4.2 Logo Sql Server

- In this project, we utilize a SQL database to store and retrieve product data, as well as store customer data. The database is running on a computer for efficient data management.

4.2 User Interface Software

4.2.1 Software User Guide

- Login System:
  + Users will be provided with login credentials (username and password) to access the system.
- Scanning Interface:
  + Utilizing a camera to scan QR codes to retrieve product information, calculate total weight, and display the corresponding price on the computer screen.
- Editing Interface:

+ The system includes functions such as adding, deleting, searching, and editing product information within the system.

o Invoice Printing:

+ Generating invoices with item details, purchase date, customer information, and pricing information.

4.2.2 User Interface and Database Integration

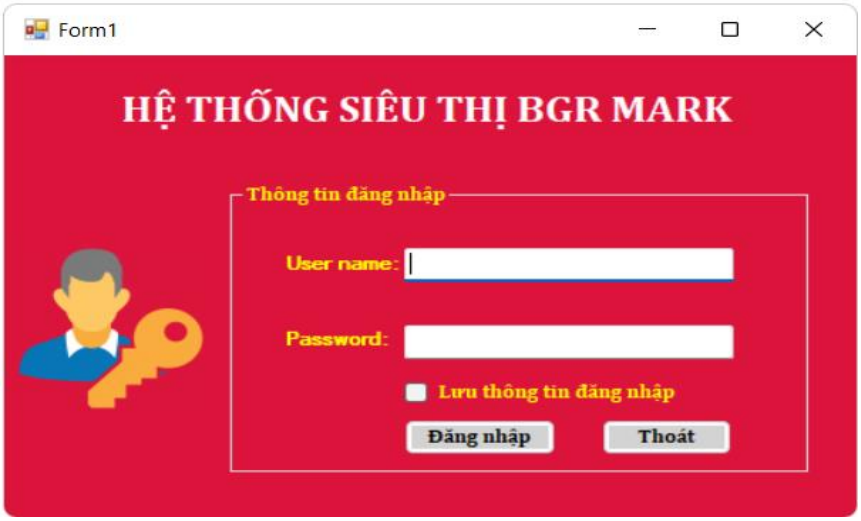


Figure 4.3 The system login interface

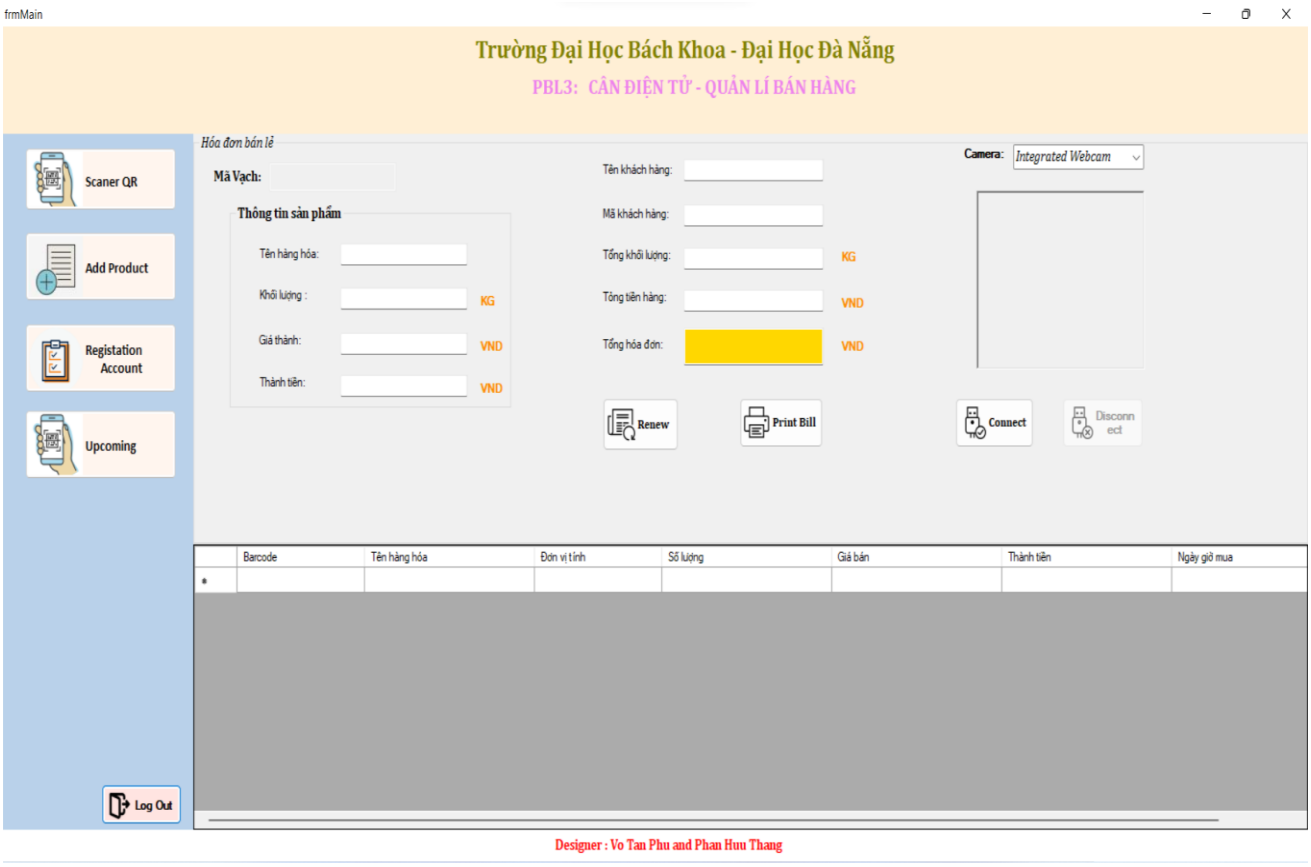


Figure 4.4 The main interface of the system for scanning QR codes

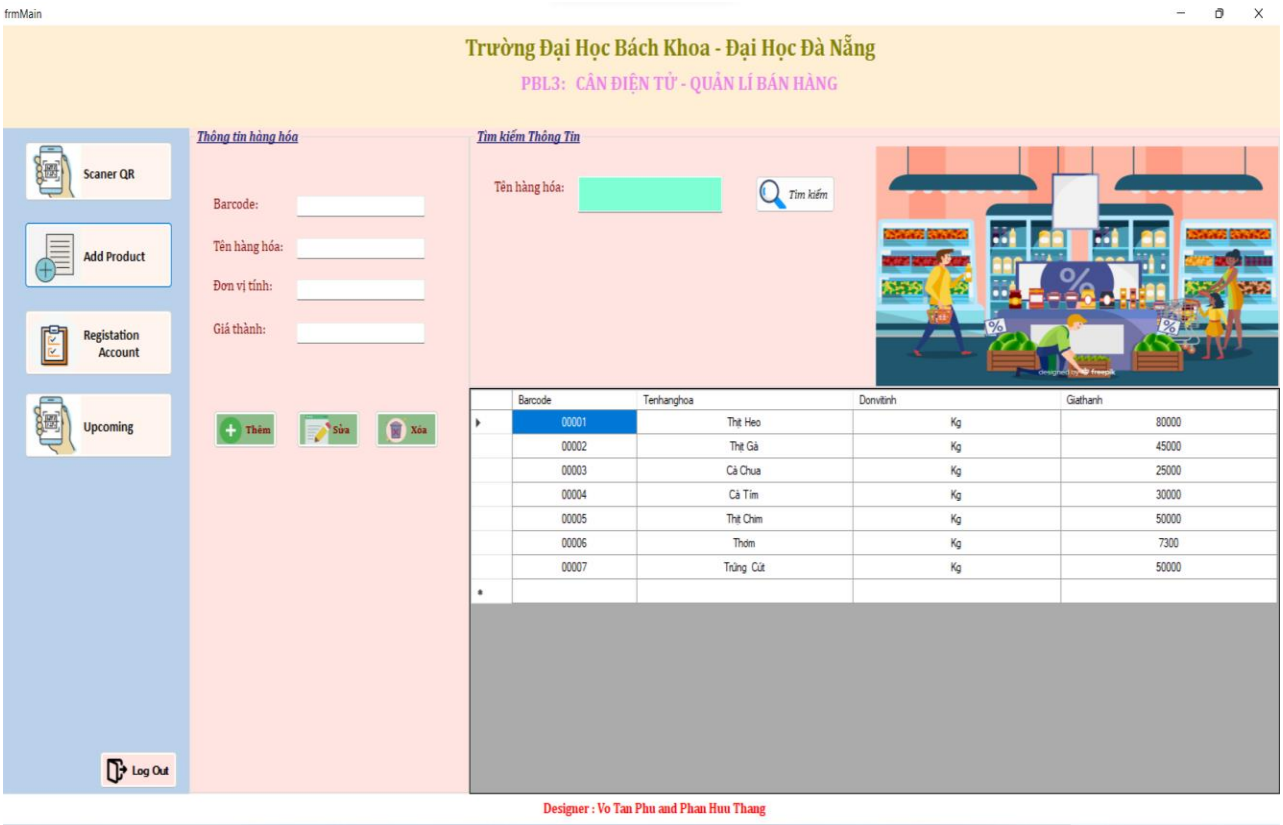


Figure 4.5 The interface for adding, deleting, and editing on the system

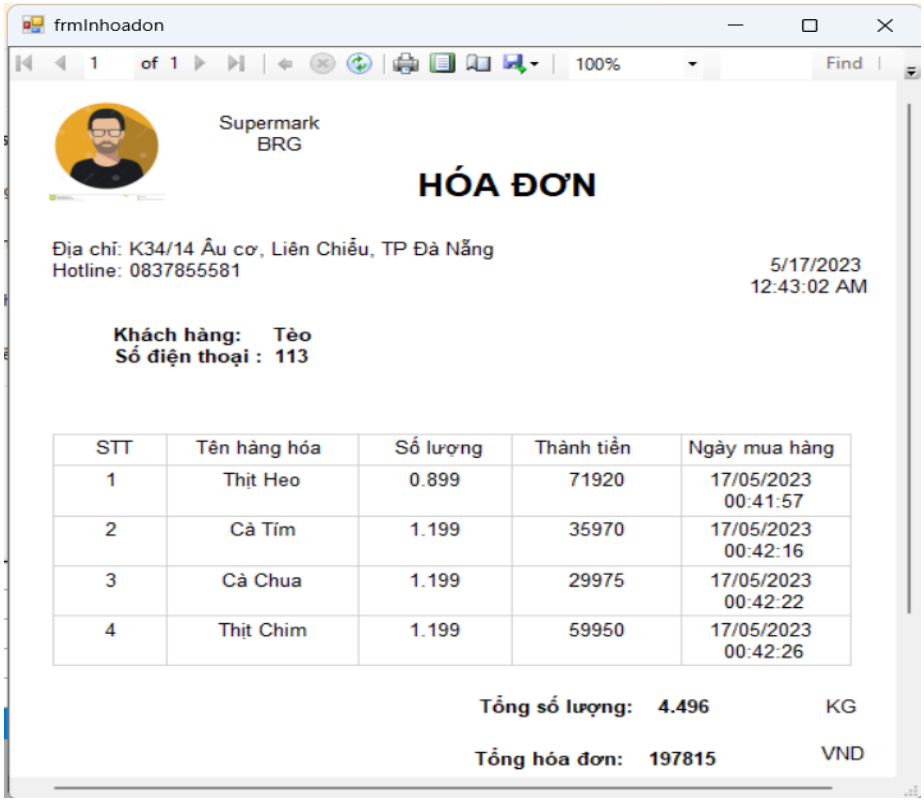


Figure 4.6 The interface for printing invoices

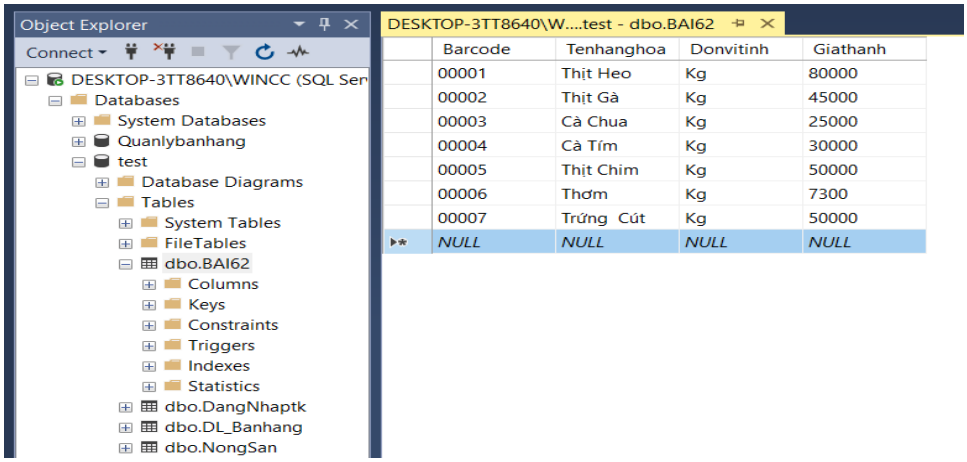


Figure 4.7 The system's product data

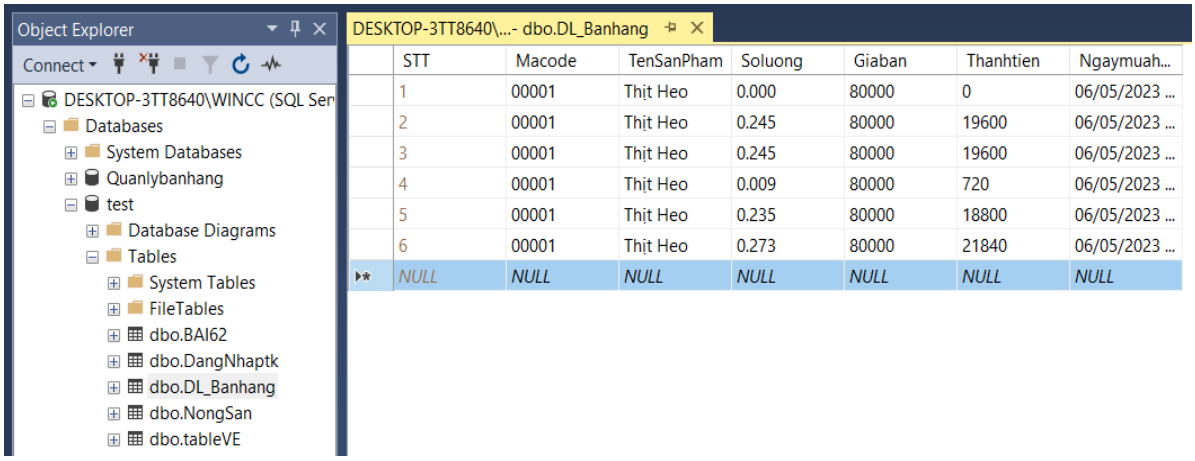


Figure 4.8 The sales data is stored in SQL Server

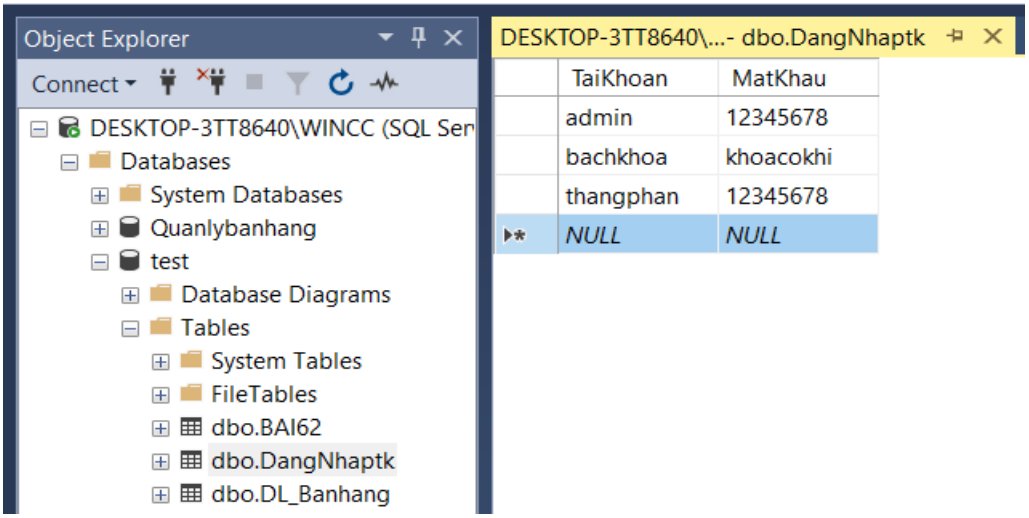


Figure 4.9 The login credentials for the system

4.2.3 Code C#

* Form Login

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace PBL3_Candientu1
{
   public partial class frmDangNhap : Form
   {
      public frmDangNhap()
      {
         InitializeComponent();
      }
      private void btnDangnhap_Click(object sender, EventArgs e)
      {

         SqlConnection Con = new SqlConnection(@"Data Source=DESKTOP-
3TT8640\WINCC;Initial Catalog=test;Integrated Security=True");   // ket noi voi CSDL
         try
         {
            Con.Open();
            string tk = txtUsername.Text;   // gan bien tk de tri xuat trong CSDL
            string mk = txtPassword.Text;
            string sql = "select *from DangNhaptk where Taikhoan='" + tk + "' and
MatKhau='" + mk + "'";
            SqlCommand cmd = new SqlCommand(sql, Con);
            SqlDataReader dta = cmd.ExecuteReader();
            if (dta.Read() == true)
            {
               MessageBox.Show("Dang nhap thanh cong");
               this.Hide();
               frmMain main = new frmMain();
               main.ShowDialog();
            }
            else
```

```
            {
                MessageBox.Show("Tai khoan hoac mat khau sai\n vui long nhap lai",
"Notify", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                txtUsername.Text = "";
                txtPassword.Text = "";
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void frmDangNhap_Load(object sender, EventArgs e)
    {

    }
  }
}
```

* Form Main (From cha)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace PBL3_Candientu1
{
  public partial class frmMain : Form
  {
    private Form activeForm;
    public frmMain()
    {
      InitializeComponent();
    }
    private void OpenChildForm(Form childForm, object btnSender)
```

```
        {
            if (activeForm != null)
                activeForm.Close();
            activeForm = childForm;
            childForm.TopLevel = false;
            childForm.FormBorderStyle = FormBorderStyle.None;
            childForm.Dock = DockStyle.Fill;
            this.panel4.Controls.Add(childForm);
            this.panel4.Tag = childForm;
            childForm.BringToFront();
            childForm.Show();
        }
        private void btnScanerQR_Click(object sender, EventArgs e)
        {
            OpenChildForm(new frmScaner(), sender);

        }
        private void btnLogOut_Click_1(object sender, EventArgs e)
        {
            DialogResult result = MessageBox.Show("ban co muon dang xuat khong", "Thong
bao", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
            if (result == DialogResult.OK)
            {
                Application.Exit();
            }
        }
        private void btnAddProduct_Click(object sender, EventArgs e)
        {
            OpenChildForm(new frmAddProduct(), sender);
        }
        private void frmMain_Load(object sender, EventArgs e)
        {
            OpenChildForm(new frmScaner(), sender);
        }
    }
}
```

* Form Scaner (Form con)

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using AForge.Video;
using AForge.Video.DirectShow;
using ZXing;
using System.IO;
using System.IO.Ports;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using System.Security.Cryptography;
using System.Net.Http.Headers;
using ZXing.QrCode.Internal;
using Microsoft.Reporting.WinForms;
namespace PBL3_Candientu1
{
   public partial class frmScaner : Form
   {
      public double Tongtienhang { get; set; }
      public double Tongkhoiluong { get; set; }
      public string TenKhachHang { get; set; }
      public string UID_Khachhang { get; set; }

      public frmScaner()
      {
         InitializeComponent();
         btnConnect.Enabled = true;     // an nut ngat ket noi
         btnDisconnect.Enabled = false;
         txtQR.Enabled = false;
      }
      private FilterInfoCollection filterInfoCollection;
      private VideoCaptureDevice captureDevice;

      private void frmScaner_Load(object sender, EventArgs e)
      {
```

```
            filterInfoCollection = new FilterInfoCollection(FilterCategory.VideoInputDevice);
            foreach (FilterInfo filterInfo in filterInfoCollection)
cboDevice.Items.Add(filterInfo.Name);
            cboDevice.SelectedIndex = 0;
            khoi_tao_bang();
        }
        private void btnConnect_Click(object sender, EventArgs e)
        {
            // thiet lap cong com giao tiep may tinh tuy vao cong com nhan ma nguoi lt thiet lap
theo y muon

            serialPort1.PortName = "COM2";
            serialPort1.BaudRate = Convert.ToInt32(9600);
            serialPort1.Open();

            // an nut ket noi di

            btnConnect.Enabled = false;
            btnDisconnect.Enabled = true;
            txtQR.Enabled = true;
            // ket noi va bat thiet bi may anh

            captureDevice = new
VideoCaptureDevice(filterInfoCollection[cboDevice.SelectedIndex].MonikerString);
            captureDevice.NewFrame += CaptureDevice_NewFrame;
            captureDevice.Start();
            timer1.Start();
            timer2.Start();
            timer3.Start();
        }
        private void CaptureDevice_NewFrame(object sender, NewFrameEventArgs
eventArgs)
        {
            pictureBox1.Image = (Bitmap)eventArgs.Frame.Clone();
        }
        private void timer1_Tick_1(object sender, EventArgs e)
        {
            if (pictureBox1.Image != null)
            {
```

```
            BarcodeReader barcodeReader = new BarcodeReader();
            Result result = barcodeReader.Decode((Bitmap)pictureBox1.Image);
            if (result != null)
            {
               txtQR.Text = result.ToString();
               timer1.Stop();
               serialPort1.Write("1");                          // gui tin hieu xuong VDK de
phat coi keu
               if (captureDevice.IsRunning)
                  captureDevice.Stop();
            }
         }

      }
      private void btnDisconnect_Click(object sender, EventArgs e)
      {
         serialPort1.Close();              // ngat giao tiep rs232
         btnConnect.Enabled = true;        // an nut ngat ket noi
         btnDisconnect.Enabled = false;
         txtQR.Text = "";
         txtQR.Enabled = false;
         timer2.Stop();
         timer3.Stop();
         if (captureDevice.IsRunning)        // Neu may anh runing thi ta off capture_device
         {
            captureDevice.Stop();
         }
         if (pictureBox1.Image != null)      // Giai phong hinh anh da scaner QR de tiep tuc
scan cac lan sau
         {
            pictureBox1.Image.Dispose();
            pictureBox1.Image = null;
         }

      }

      private void timer2_Tick(object sender, EventArgs e)
      {
         string tam = serialPort1.ReadExisting();
```

```
            if (tam != "")
            {
                txtKhoiluong.Text = tam;
            }

        }
        string strCon = @"Data Source=DESKTOP-3TT8640\WINCC;Initial
Catalog=test;Integrated Security=True";
        SqlConnection sqlCon = null;
        double sum = 0;
        string donvi_tam;
        int tam1 = 0;
        private void txtQR_TextChanged_1(object sender, EventArgs e)
        {
            if (txtQR.Text != "")
            {
                if (sqlCon == null)                                    // rong thi tao va
dong thi mo
                {
                    sqlCon = new SqlConnection(strCon);
                }
                if (sqlCon.State == ConnectionState.Closed)
                {
                    sqlCon.Open();
                }
                string maBarcode = txtQR.Text.Trim();                  // doi tuong
thuc thi tri van lay csdl
                SqlCommand sqlCmd = new SqlCommand();
                sqlCmd.CommandType = CommandType.Text;
                using (SqlCommand cmd = new SqlCommand("select * from BAI62 where
Barcode ='" + maBarcode + "'", sqlCon))        // gui truy van vao ket noi
                {
                    using (SqlDataReader reader = cmd.ExecuteReader())      // thuc thidoc du lieu
                    {
                        while (reader.Read())
                        {
                            string tenhanghoa = reader.GetString(1);
                            string donvitinh = reader.GetString(2);
                            donvi_tam = donvitinh;
```

```
                    string giaThanh = reader.GetString(3);
                    txtTenhanghoa.Text = tenhanghoa;
                    txtGiathanh.Text = giaThanh;

                    int a = Convert.ToInt32(reader.GetString(3));                // bien gia thanh
                    double b = Convert.ToDouble(txtKhoiluong.Text);              // bien khoi
luong
                    Tongtienhang = Tongtienhang + a * b;
                    Tongkhoiluong = Tongkhoiluong + b;
                    sum = a * b;

                    txtThanhtien.Text = Convert.ToString(sum);
                    txtTongkhoiluong.Text = Convert.ToString(Tongkhoiluong);
                    txtTongtienhang.Text = Convert.ToString(Tongtienhang);
                    txtTonghoadon.Text = Convert.ToString(Tongtienhang);
                    Add_dulieu();
                }
                reader.Close();
            }
        }
        if (dataGridView1.Rows.Count - 1 > 0 && tam1 == 1)
        {
            using (SqlCommand sqlCmd2 = new SqlCommand("Insert into
DL_Banhang(Macode,TenSanPham,Soluong,Giaban,Thanhtien,Ngaymuahang)
values(@code,@tensanpham,@soluong,@giaban,@thanhtien,@ngaymuahang)", sqlCon))
            {
                sqlCmd2.CommandType = CommandType.Text;
                sqlCmd2.Parameters.AddWithValue("@code", txtQR.Text);
                sqlCmd2.Parameters.AddWithValue("@tensanpham", txtTenhanghoa.Text);
                sqlCmd2.Parameters.AddWithValue("@soluong", txtKhoiluong.Text);
                sqlCmd2.Parameters.AddWithValue("@giaban", txtGiathanh.Text);
                sqlCmd2.Parameters.AddWithValue("@thanhtien", txtThanhtien.Text);
                sqlCmd2.Parameters.AddWithValue("@ngaymuahang", lblClock.Text);
                sqlCmd2.ExecuteNonQuery();
                tam1 = 0;
            }
        }
    }
```

```
    private void khoi_tao_bang()
    {
        dataGridView1.AutoGenerateColumns = false;
        const int NumberColumn = 7;
        dataGridView1.ColumnCount = NumberColumn;
        string[] list = new string[NumberColumn] { "Barcode", "Tenhanghoa", "Donvitinh",
"Soluong", "Giaban", "Thanhtien", "Ngaygiomua" };
        string[] header_name = new string[NumberColumn] { "Barcode", "Tên hàng hóa",
"Đơn vị tính", "Số lượng", "Giá bán", "Thành tiền", "Ngày giờ mua" };
        int[] width_col = new int[NumberColumn] { 150, 200, 150, 200, 200, 200, 250 };

        for (int i = 0; i < NumberColumn; i++)
        {
            dataGridView1.Columns[i].Name = list[i];
            dataGridView1.Columns[i].DataPropertyName = list[i];
            dataGridView1.Columns[i].HeaderText = header_name[i];
            dataGridView1.Columns[i].Width = width_col[i];
            dataGridView1.Columns[i].ValueType = typeof(String);
            dataGridView1.Columns[i].Visible = true;
        }
    }

    private void timer3_Tick(object sender, EventArgs e)
    {
        DateTime currentDateTime = DateTime.Now;
        lblClock.Text = currentDateTime.ToString("dd/MM/yyyy HH:mm:ss");
    }

    private void btnXoa_Click(object sender, EventArgs e)
    {
        sum = 0;
        Tongkhoiluong = 0;
        Tongtienhang = 0;
        tam1 = 0;
        txtTenKH.Text = "";
        txtMaKH.Text = "";
        dataGridView1.Rows.Clear();
    }
```

```
public void Add_dulieu()
{
   DataGridViewRow row = new DataGridViewRow();
   row.CreateCells(dataGridView1);
   row.Cells[0].Value = txtQR.Text;
   row.Cells[1].Value = txtTenhanghoa.Text;
   row.Cells[2].Value = donvi_tam;
   row.Cells[3].Value = txtKhoiluong.Text;
   row.Cells[4].Value = txtGiathanh.Text;
   row.Cells[5].Value = txtThanhtien.Text;
   row.Cells[6].Value = lblClock.Text;
   dataGridView1.Rows.Add(row);
   tam1++;

}
ReportDataSource rs = new ReportDataSource();
private void btnInhoadon_Click(object sender, EventArgs e)
{
   if(txtMaKH.Text.Length > 0 && txtTenKH.Text.Length >0)
   {
      TenKhachHang = txtTenKH.Text;
      UID_Khachhang = txtMaKH.Text;
      List<Dulieu1> list = new List<Dulieu1>();
      list.Clear();

      for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
      {
         Dulieu1 dulieu1 = new Dulieu1
         {
            Tenhanghoa = dataGridView1.Rows[i].Cells[1].Value.ToString(),
            Soluong = dataGridView1.Rows[i].Cells[3].Value.ToString(),
            Thanhtien = dataGridView1.Rows[i].Cells[5].Value.ToString(),
            Thoigianmua = dataGridView1.Rows[i].Cells[6].Value.ToString()
         };
         list.Add(dulieu1);
      }

      rs.Name = "DataSet1";
      rs.Value = list;
```

```
            frmInhoadon frm2 = new frmInhoadon(this);
            frm2.reportViewer1.LocalReport.DataSources.Clear();
            frm2.reportViewer1.LocalReport.DataSources.Add(rs);
            frm2.reportViewer1.LocalReport.ReportEmbeddedResource =
"PBL3_Candientu1.Report1.rdlc";
            frm2.ShowDialog();
        }
        else
        {
            MessageBox.Show("Vui long nhap ten KH", "Notify", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }

    }
  }
}
```

* Form Add Product (Form con)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Microsoft.ReportingServices.Diagnostics.Internal;

namespace PBL3_Candientu1
{
  public partial class frmAddProduct : Form
  {
    SqlConnection connection;
    SqlCommand command;
    string sqlConn = @"Data Source=DESKTOP-3TT8640\WINCC;Initial
Catalog=test;Integrated Security=True";
    SqlDataAdapter adapter = new SqlDataAdapter();
    DataTable table = new DataTable();              // day du lieu xuong table
```

```
    public frmAddProduct()
    {
        InitializeComponent();
    }

    private void frmAddProduct_Load(object sender, EventArgs e)
    {
        connection= new SqlConnection(sqlConn);
        connection.Open();
        khoitaobang();
    }
    private void khoitaobang()
    {
        command = connection.CreateCommand();
        command.CommandText = "select * from BAI62";
        adapter.SelectCommand= command;
        table.Clear();
        adapter.Fill(table);
        dataGridView.DataSource= table;
        dataGridView.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        dataGridView.Columns[0].Width = 150;
        dataGridView.Columns[1].Width = 260;
        dataGridView.Columns[2].Width = 250;
        dataGridView.Columns[3].Width = 260;
    }

    private void btnThem_Click(object sender, EventArgs e)
    {
        command = connection.CreateCommand();
        command.CommandText = "Insert into BAI62 values(@barcode, @tenhanghoa,
@donvitinh, @giathanh)";
        command.Parameters.AddWithValue("@barcode", txtBarcode.Text);
        command.Parameters.AddWithValue("@tenhanghoa", txtTenhanghoa.Text);
        command.Parameters.AddWithValue("@donvitinh", txtDonvitinh.Text);
        command.Parameters.AddWithValue("@giathanh", txtGiathanh.Text);
        command.ExecuteNonQuery();
        khoitaobang();
    }
```

```
        private void btnSua_Click(object sender, EventArgs e)
        {
            command = connection.CreateCommand();
            command.CommandText = "UPDATE BAI62 SET Barcode =
'"+txtBarcode.Text+"',Tenhanghoa = N'"+txtTenhanghoa.Text+"',Donvitinh =
'"+txtDonvitinh.Text+"',Giathanh = '"+txtGiathanh.Text+"' where Barcode =
'"+txtBarcode.Text+"'";

            command.ExecuteNonQuery();
            khoitaobang();
        }

        private void btndelete_Click(object sender, EventArgs e)
        {
            command = connection.CreateCommand();
            command.CommandText = "delete from BAI62  where Barcode=(@barcode)";
            command.Parameters.AddWithValue("@barcode", txtBarcode.Text);
            command.ExecuteNonQuery();
            khoitaobang();
        }

        private void dataGridView_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            txtBarcode.ReadOnly = true;
            int i;
            i = dataGridView.CurrentRow.Index;
            txtBarcode.Text = dataGridView.Rows[i].Cells[0].Value.ToString();
            txtTenhanghoa.Text = dataGridView.Rows[i].Cells[1].Value.ToString();
            txtDonvitinh.Text = dataGridView.Rows[i].Cells[2].Value.ToString();
            txtGiathanh.Text = dataGridView.Rows[i].Cells[3].Value.ToString();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            command = connection.CreateCommand();
            command.CommandText = "select * from BAI62 where Tenhanghoa LIKE '%' +
@tenhanghoa + '%'";
            command.Parameters.AddWithValue("@tenhanghoa", txtTen_S.Text);
```

```
            command.ExecuteNonQuery();
            SqlDataReader dapter1 = command.ExecuteReader();
            DataTable table1 = new DataTable();
            table1.Load(dapter1);
            dataGridView.DataSource= table1;
        }
    }
}
```

* Form In Hóa Đơn

```
using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace PBL3_Candientu1
{
    public partial class frmInhoadon : Form
    {
        private frmScaner frmscaner;
        public frmInhoadon(frmScaner frmscaner)
        {
            InitializeComponent();
            this.frmscaner = frmscaner;
        }
        public void frmInhoadon_Load(object sender, EventArgs e)
        {
            double tongtienhang = this.frmscaner.Tongtienhang;
            double tongkhoiluong = this.frmscaner.Tongkhoiluong;
            string tenkhachhang = this.frmscaner.TenKhachHang;
            string sodienthoai = this.frmscaner.UID_Khachhang;
            ReportParameter Tongtienhang = new ReportParameter("Tongtienhang",
tongtienhang.ToString());
            this.reportViewer1.LocalReport.SetParameters(new ReportParameter[] {
Tongtienhang });
```

```
            ReportParameter Tongkhoiluong = new ReportParameter("Tongkhoiluong",
tongkhoiluong.ToString());
            this.reportViewer1.LocalReport.SetParameters(new ReportParameter[] {
Tongkhoiluong });
            ReportParameter TenKhachHang = new ReportParameter("TenKhachHang",
tenkhachhang);
            this.reportViewer1.LocalReport.SetParameters(new ReportParameter[] {
TenKhachHang });
            ReportParameter UID_Khachhang = new ReportParameter("UID_Khachhang",
sodienthoai);
            this.reportViewer1.LocalReport.SetParameters(new ReportParameter[] {
UID_Khachhang });

            this.reportViewer1.RefreshReport();
        }
    }
}
```

* Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace PBL3_Candientu1
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new frmDangNhap());
        }
    }
}
```

CHAPTER 5 LESSON LEARNED AND CONCLUSION

5.1 Overall Evaluation

- The topic of the Electronic Scale and Sales Management System is a product that our team has developed, which is not only a model but also intended to be used in real life. Currently, there are many types of integrated barcode scanning scales on the market, but they are quite expensive. Therefore, our team has made efforts to utilize available components to implement the project at a lower cost compared to other electronic scales currently on the market. And we aim to continuously improve the developed product.
- During the project development process, we encountered several difficulties in printing and circuit fabrication. However, the etching process was relatively easy. The process of drilling holes and soldering components did not pose many challenges

5.2 Advantages and disadvantages of the product

a) Advantages

+ Quick QR code scanning utilizing the built-in camera of the laptop
+ User-friendly interface
+ Integration of multiple functions
+ Simple installation, compact model
+ Easy to use

b) Disadvantages

+ Lower quality
+ Lower accuracy
+ Interface requires improvement

5.3 Future Development Directions

+ Strive to improve the accuracy of the electronic scale.
+ Optimize the program algorithm.
+ Optimize the design of the model.
+ Improve the quality of the product by using sensors with higher accuracy.

REFERENCE

[1] Document “Giáo trình vi điều khiển PIC” – Đặng Phước Vinh & Võ Như Thành, published in 2021.
[2] Document “Giáo trình kĩ thuật vi điều khiển – Đặng Phước Vinh, published in 2019
[3] Datasheet HX711 - Access link:https://pdf1.alldatasheet.com/datasheetpdf/view/1132222/AVIA/HX711.html . Access on 23/3/2023.
[4].Datasheet PIC16F877A – Access link: https://pdf1.alldatasheet.com/datasheet-pdf/view/82338/MICROCHIP/PIC16F877A.html . Access on 23/3/2023
[5] Document “Xử lí ảnh” – Nguyen Dinh Cuong. Access link: https://bom.so/z3gR3k and access on 22/3/2023