Secure user interface chat over RF

Mahmoud Mostafa
PE1
r0997618

- Project Description

The RF-based secure wireless communication system allows for reliable and versatile wireless communication using the XC888 microcontroller, an LCD display, a PS/2 keyboard, and an NRF24L01 transceiver. This project is designed to facilitate wireless data transfer between two systems, complete with additional features such as battery percentage monitoring and clock display.

- System Overview

Microcontroller: XC888

Display: 20x4 LCD display

Input: PS/2 keyboard

Wireless Communication: NRF24L01 transceiver

Power Supply: 7.4V LiPo battery

Key Features

- Wireless Communication:

Utilizes the NRF24L01 transceiver for robust and efficient wireless data transfer.

Supports two-way communication between systems, enabling both transmission and reception of messages.

User Interface:

An LCD display is used to show messages, battery percentage, and the current time.

A PS/2 keyboard is used for user input, allowing users to type messages directly into the system.

Message Input and Display:

Users can enter messages up to 32 characters in length.

The system displays the entered message on the LCD screen.

The clock feature shows the current time on the display.

Battery Monitoring:

The system monitors the battery percentage, providing users with real-time information about the power status.

The battery percentage is displayed on the LCD, ensuring users are aware of the remaining battery life.

Operation

System Start-Up:

When the system is powered on, it initializes the microcontroller, LCD display, PS/2 keyboard, and NRF24L01 transceiver.

Message Entry: welcome the user and let him choose wether he want to chat or see the time or the battery percentage

Users can type a message using the PS/2 keyboard.

The message is displayed on the LCD as it is being typed.

The maximum message length is 32 characters.

Wireless Communication:

Once the message is entered, it can be transmitted wirelessly to another system using the NRF24L01 transceiver.

The receiving system displays the incoming message on its LCD.

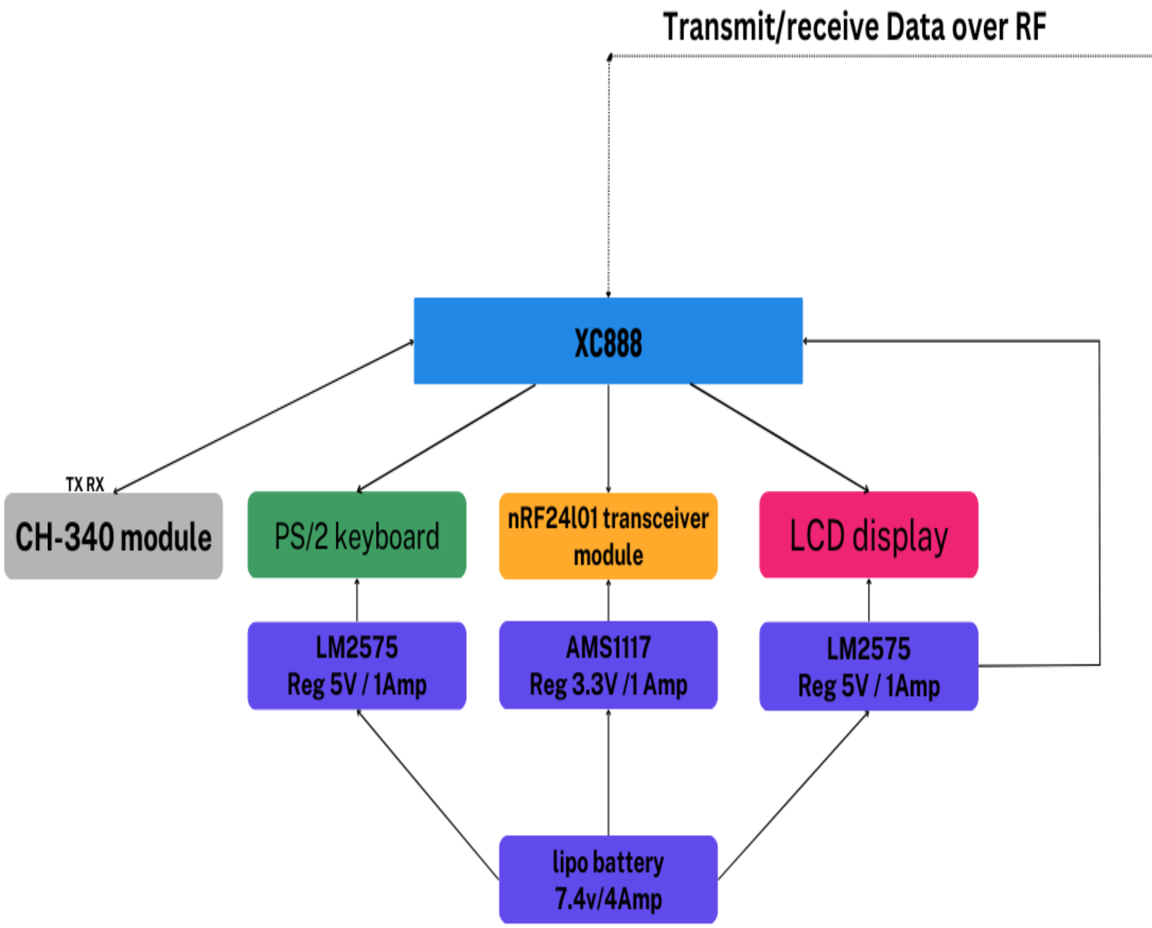
Battery and Clock Display:

Throughout the operation, the LCD continuously displays the battery percentage and the current time.

This ensures users have constant visibility of the system status.

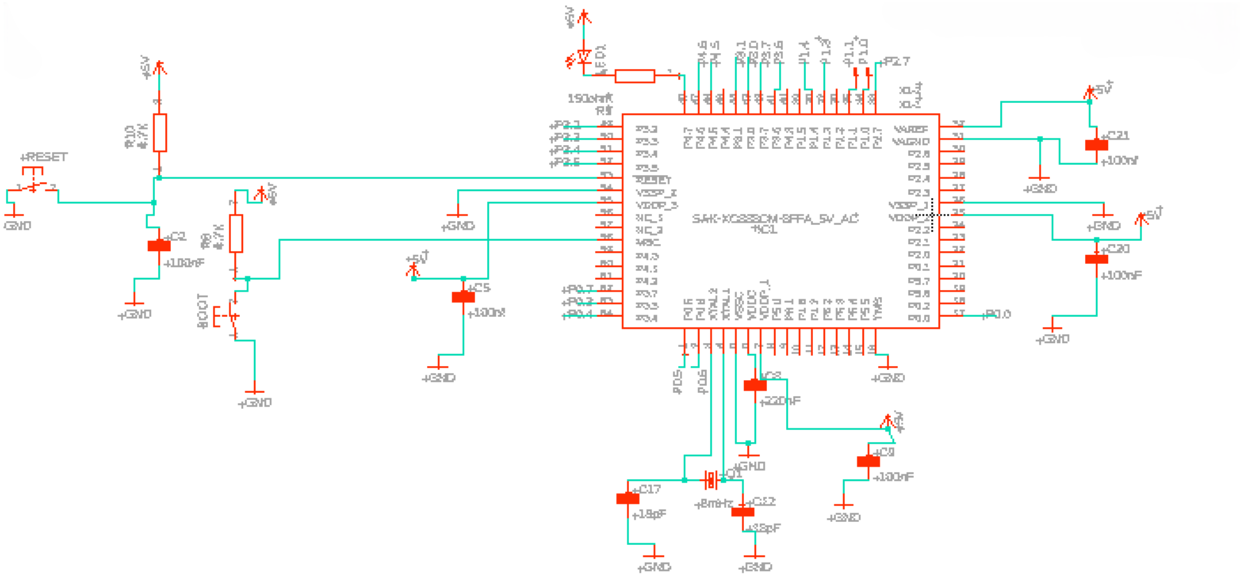
This project demonstrates a practical and versatile wireless communication system with additional functionality for battery monitoring and time display, providing a comprehensive solution for reliable and user-friendly wireless communication.

ii. Block Diagram Hardware



ii. Schematic

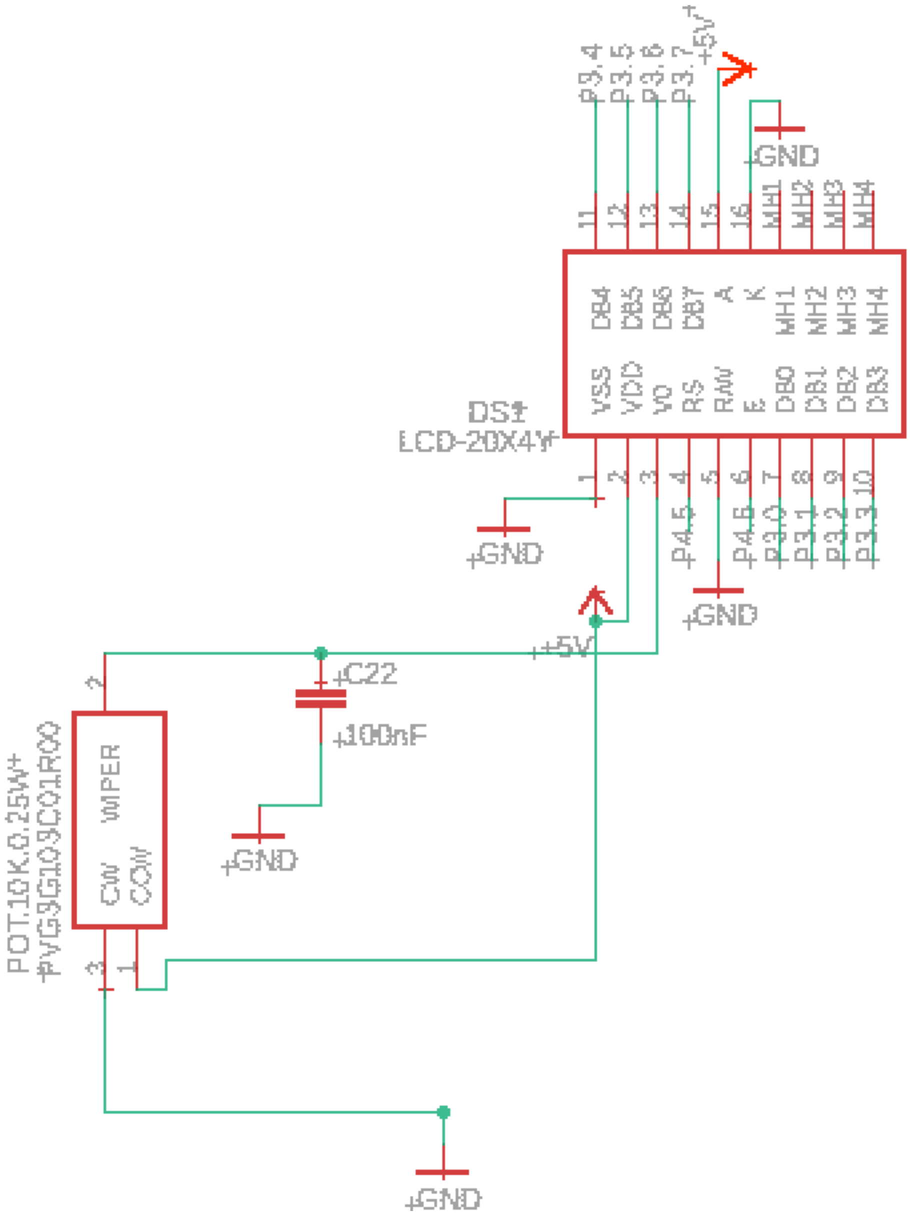
- Xc888



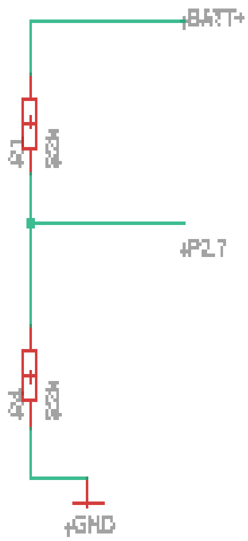
- LCD Display

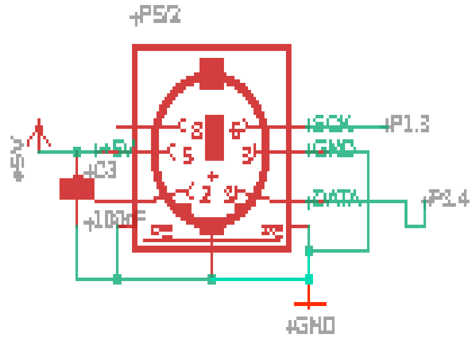
Data pins: P3.0 to P3.7

Control pins: P4.5 and P4.6



- PS/2 keyboard and battery monitor





- NRF24L01+ module with 5x 74LV1t3GV Voltage Translator

Pins: Port 0

CE= P0.7 OUTPUT

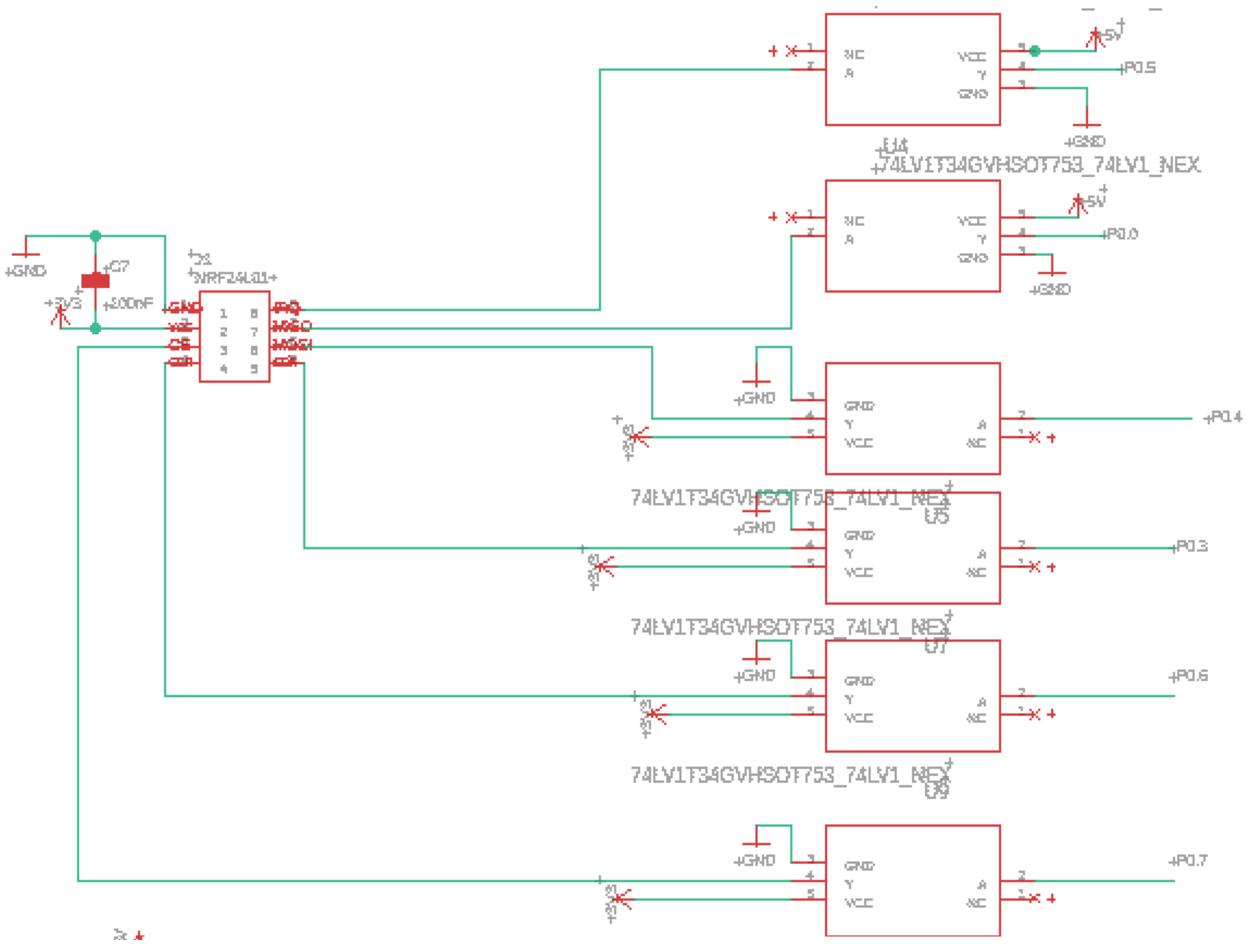
CSN= P0.6 OUTPUT

IRQ= P05 INPUT

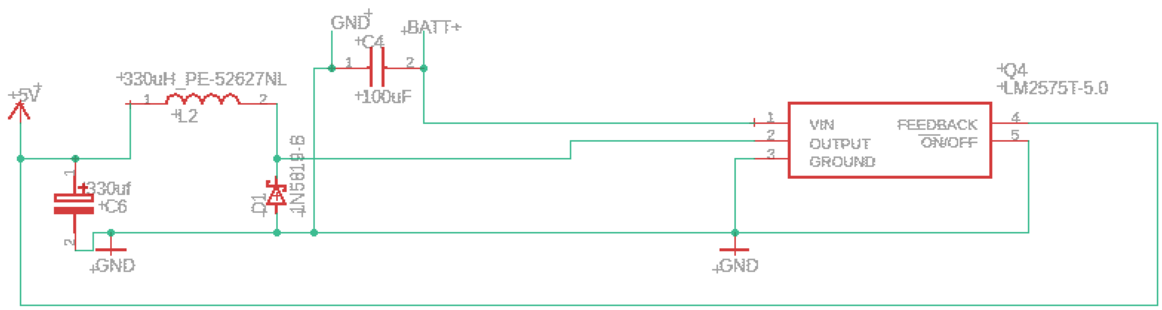
MOSI= P0.4 OUTPUT

SCK= P0.3 OUTPUT

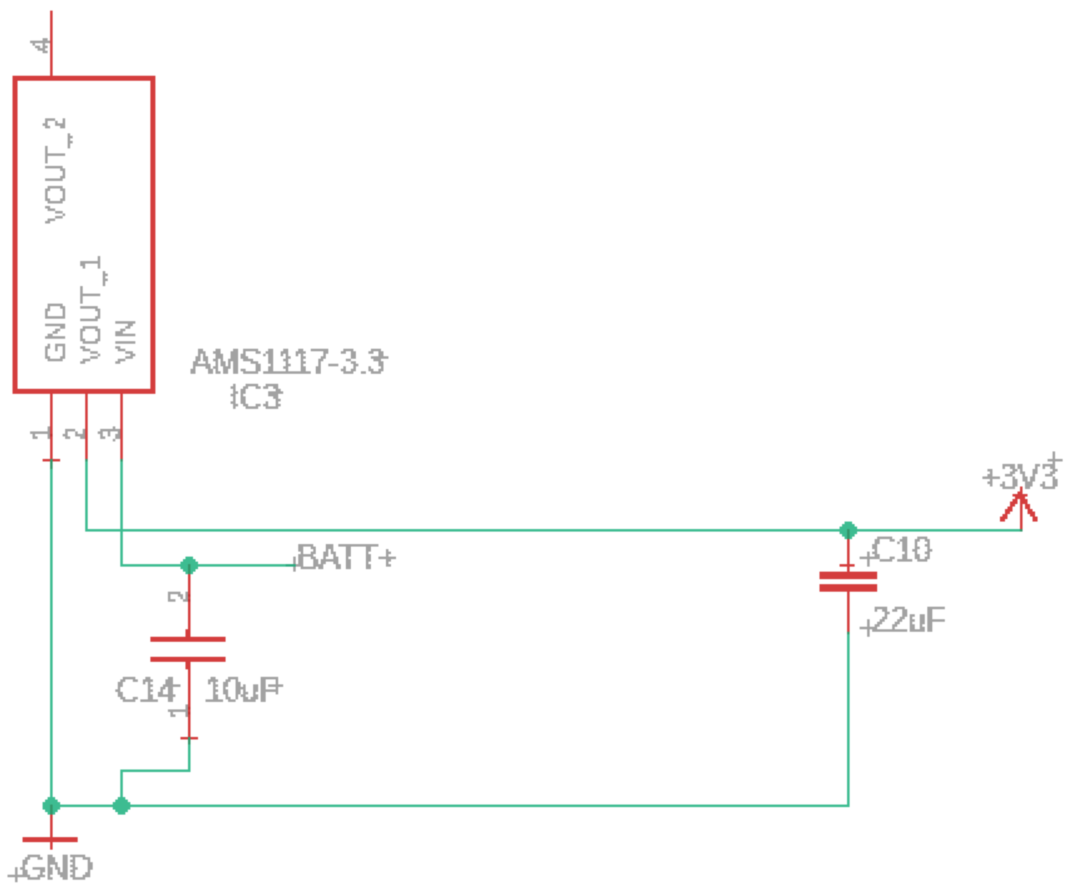
MISO= p0.0 INPUT



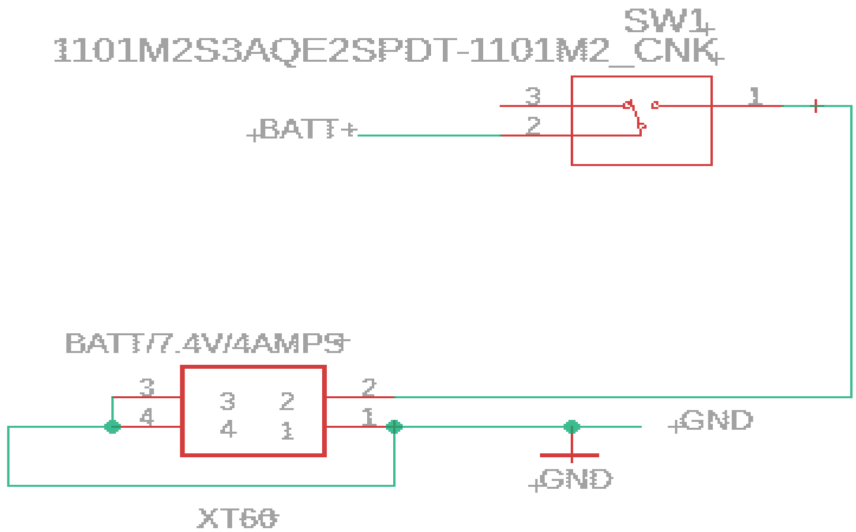
- LM2575 5V-1A Voltage Regulator



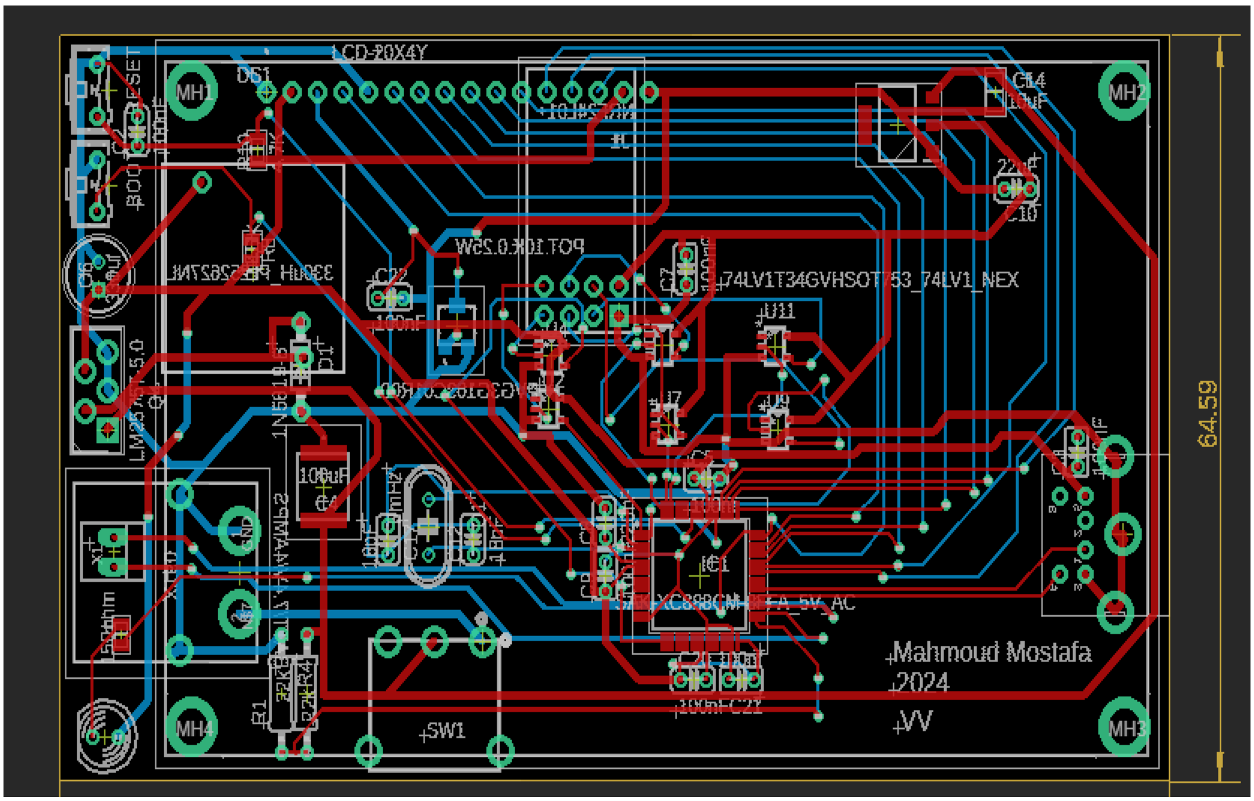
- AMS1117-3.3V -1A voltage Regulator
  Note : dropdown voltage 1.1V



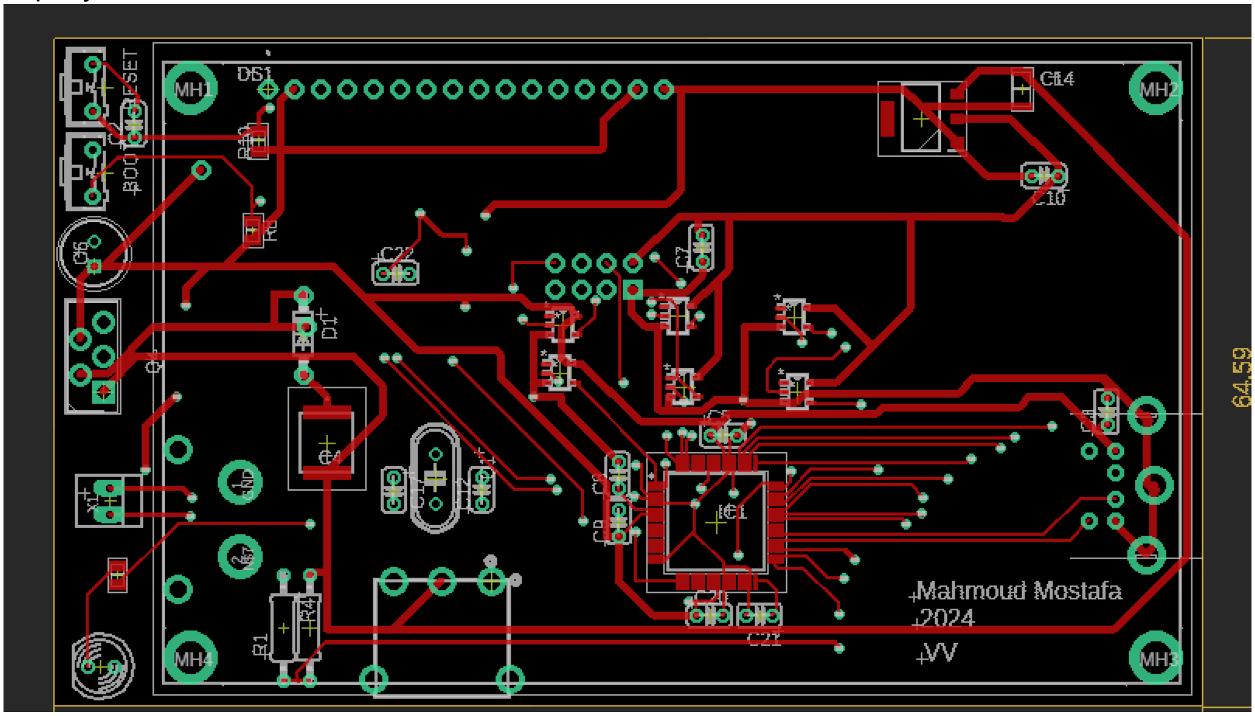
- 7.4V 6000mA Li-poly Battery with slide switch
  Note battery input is XT60


SW1
1101M2S3AQE2SPDT-1101M2_CNK
BATT+
BATT/7.4V/4AMPS
XT60
GND
GND

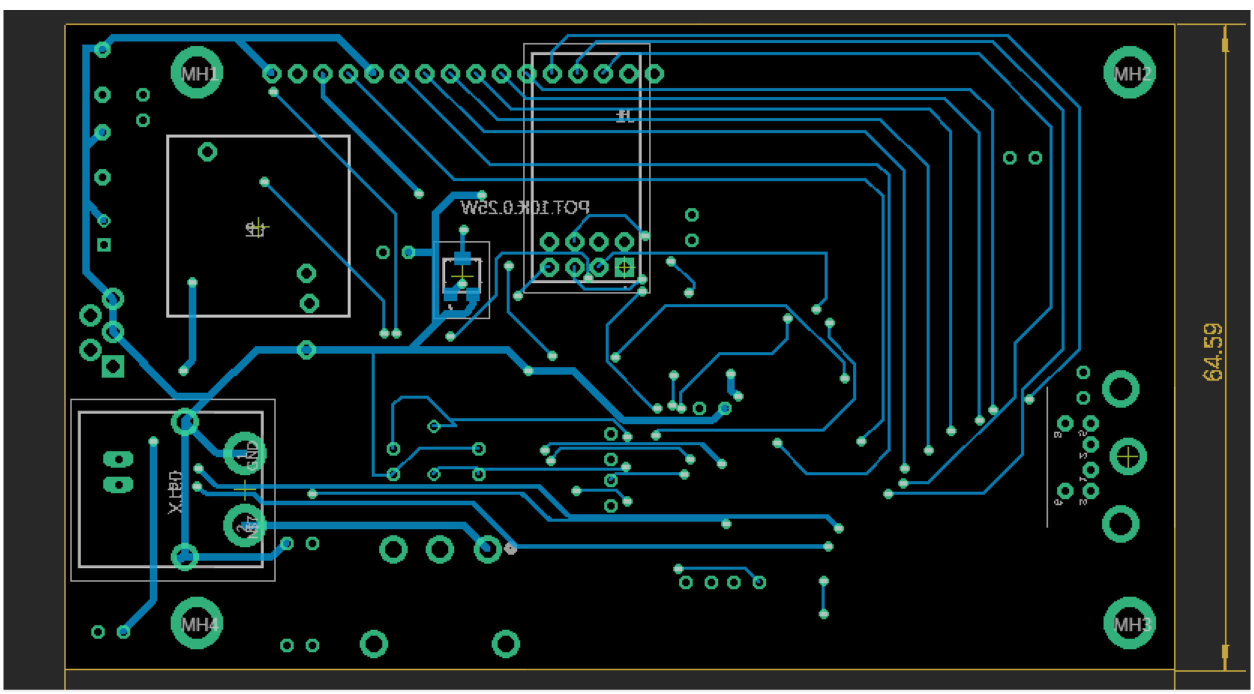
- PCB layout

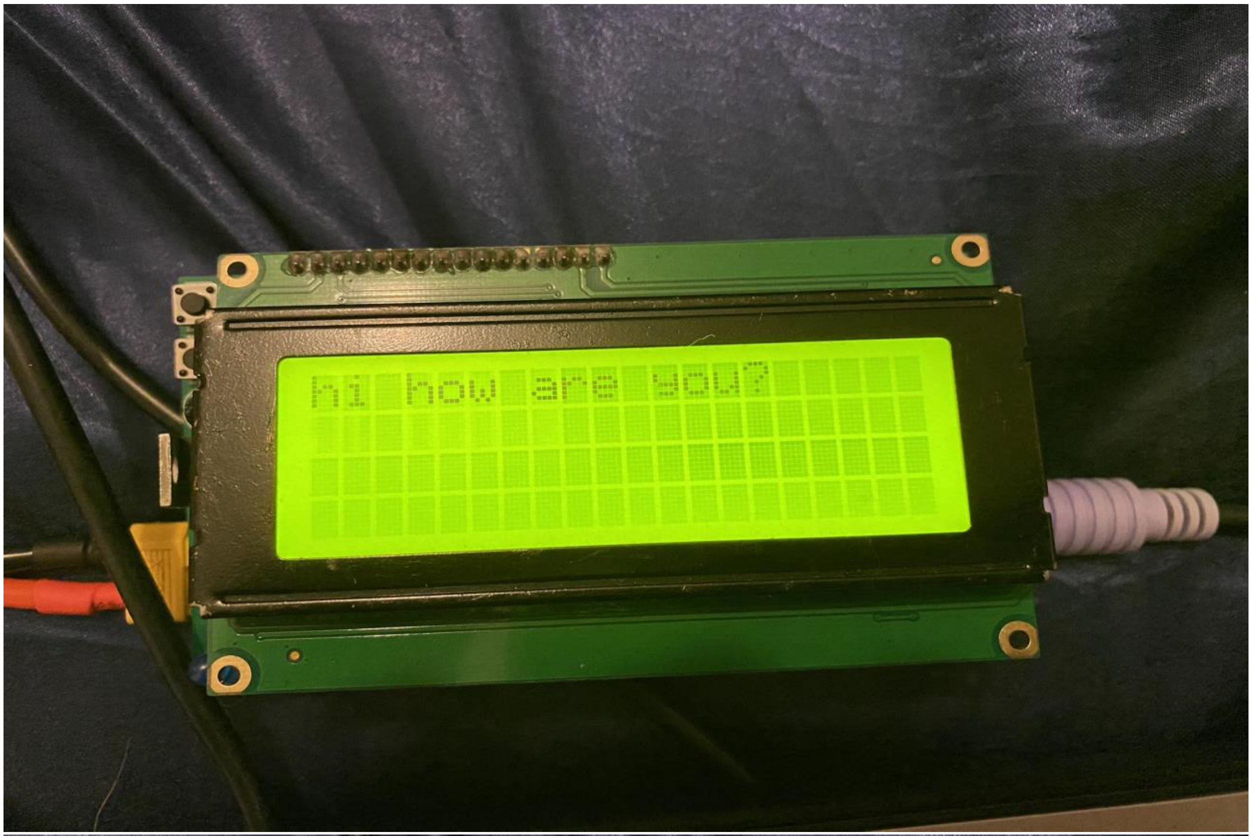

LCD-20X4Y
Mahmoud Mostafa
2024
VV
64.59
SW1
MH1
MH2
MH3
MH4

Top layer:

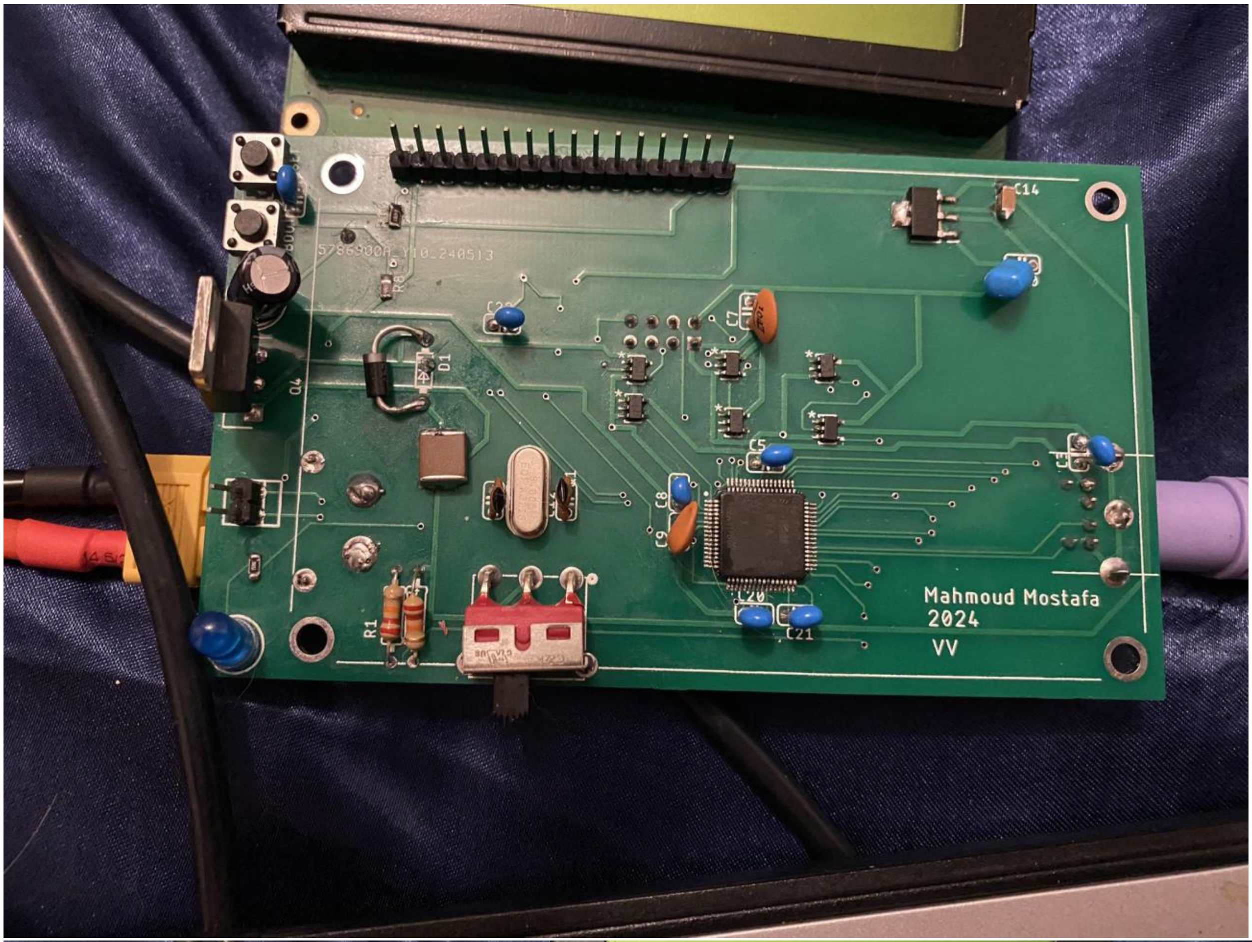


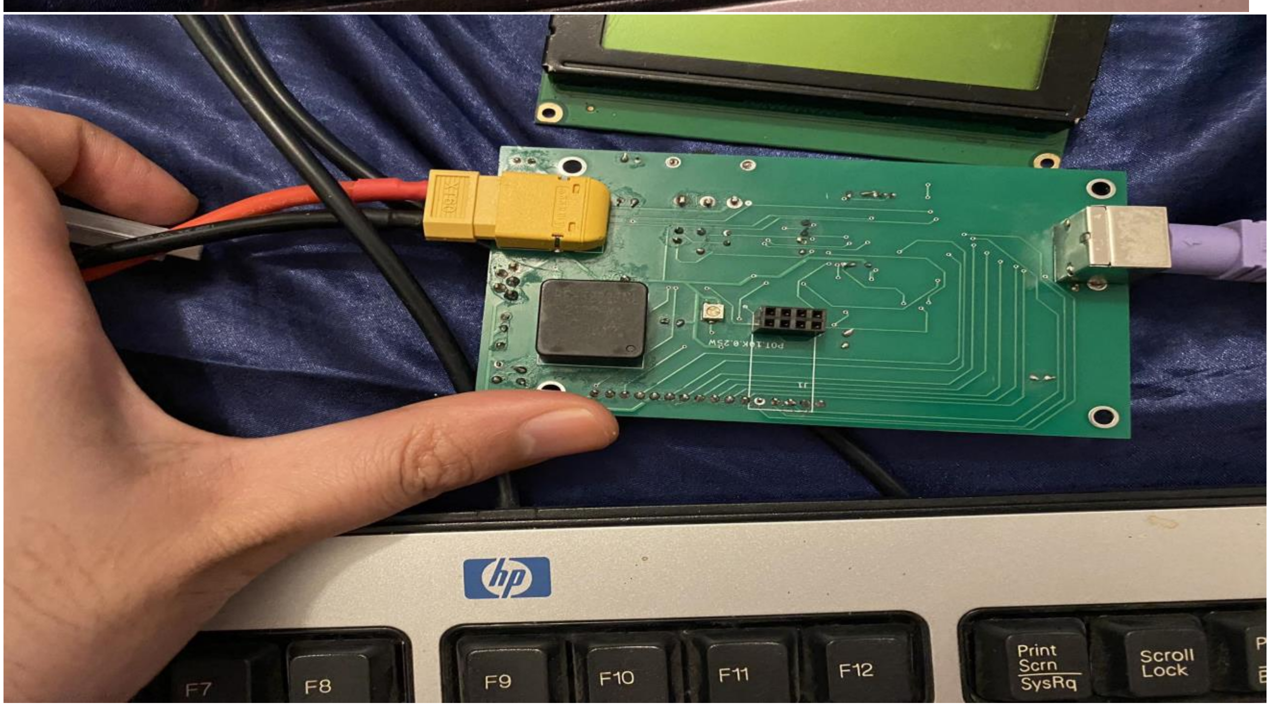
Bottom layer:




hi how are you?


1.Chat
2.Battery
3.clock
Press 1,2 or3:
F5
F6
F7
F8
F9
F10
F11
F12
Print Scrn SysRq
Scroll Lock
Pause Break
Insert
Home
Page Up
Num Lock


Mahmoud Mostafa
2024
VV
C14
C5
C8
C9
C21
D1
R1
Q4


F7
F8
F9
F10
F11
F12
Print
Scrn
SysRq
Scroll
Lock

iii. Software structure

- PS/2 keyboard

The first part of the software fetches the SCAN CODE from the PS/2 keyboard SCAN CODE is a code which is send by a keyboard to a computer whenever a key is pressed. All the keys have different scan code. So it is possible to detect which key is pressed on the keyboard by looking at the scan code the keyboard has sent to the computer.

The scan code, then, needs to be converted to equivalent ASCII code. the LCD Screen that i have used here can display alphabets only when the equivalent ASCII code is sent to the LCD screen.

The SCAN CODE actually has two parts- MAKE CODE and BREAK CODE. Both these codes combine to form the SCAN CODE. The MAKE CODE is generated by the KEYBOARD ENCODER when an USER just presses the key and the BREAK CODE is generated when the USER release the key. For example, when the key ESC is pressed the MAKE CODE generated by the KEYBOARD ENCODER is 76H and sends to the computer. When the pressed ESC key is released, the BREAK CODE is generated by the KEYBOARD ENCODER is F0H, 76H and then again this code is send to the computer. In this way the KEYBOARD ENCODER generates code for all the keys and sends to the xc888

The keyboard communication protocol is a serial communication protocol. The keyboard sends data **11 bits** in the following order:-

– 1 start bit (always 0);

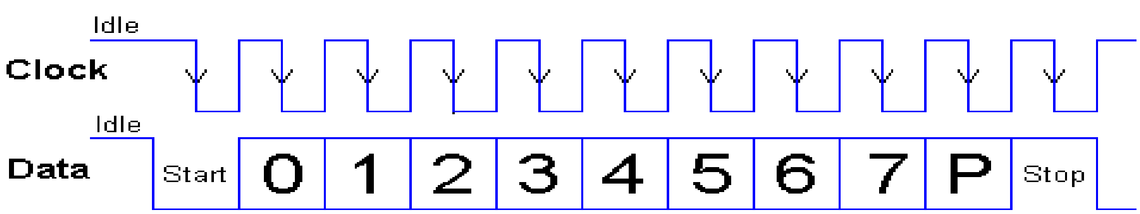
– 8 data bits (LSB first);

– 1 parity bit (if number of ones is even, then parity bit = 1);

– 1 stop bit (always 1).

The data are sent by the keyboard serially using the data line. The clock line synchronizes the data line during communication. The clock is produced by the keyboard itself. A diagram for both the line status has been given below.

the keyboard sends one byte of data at a time. (From 0 to 7 bits contribute to form one byte in the above



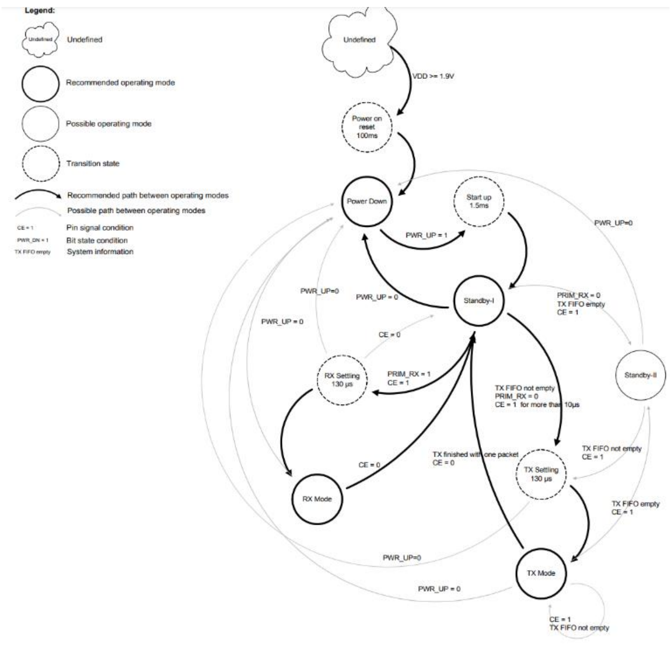
picture). The program detects each bit of one byte **data at the rising edge** of the each clock pulse.

- nRF24L01+ module

features:

- nRF24L01+ supports air data rates of 250 kbps, 1 Mbps, and 2 Mbps
- 2.4GHz RF channel
- Ultra low power operation
- 1.9v to 3.6v power supply
- SPI interface
- embedded baseband protocol engine (Enhanced ShockBurst™) is based on packet communication and supports various modes

modes and packet structure:



Preamble 1 byte	Address 3-5 byte	Packet Control Field 9 bit	Payload 0 - 32 byte	CRC 1-2 byte

Figure 4. An Enhanced ShockBurst™ packet with payload (0-32 bytes)

6.1.6 Operational modes configuration

The following table (Table 15.) describes how to configure the operational modes.

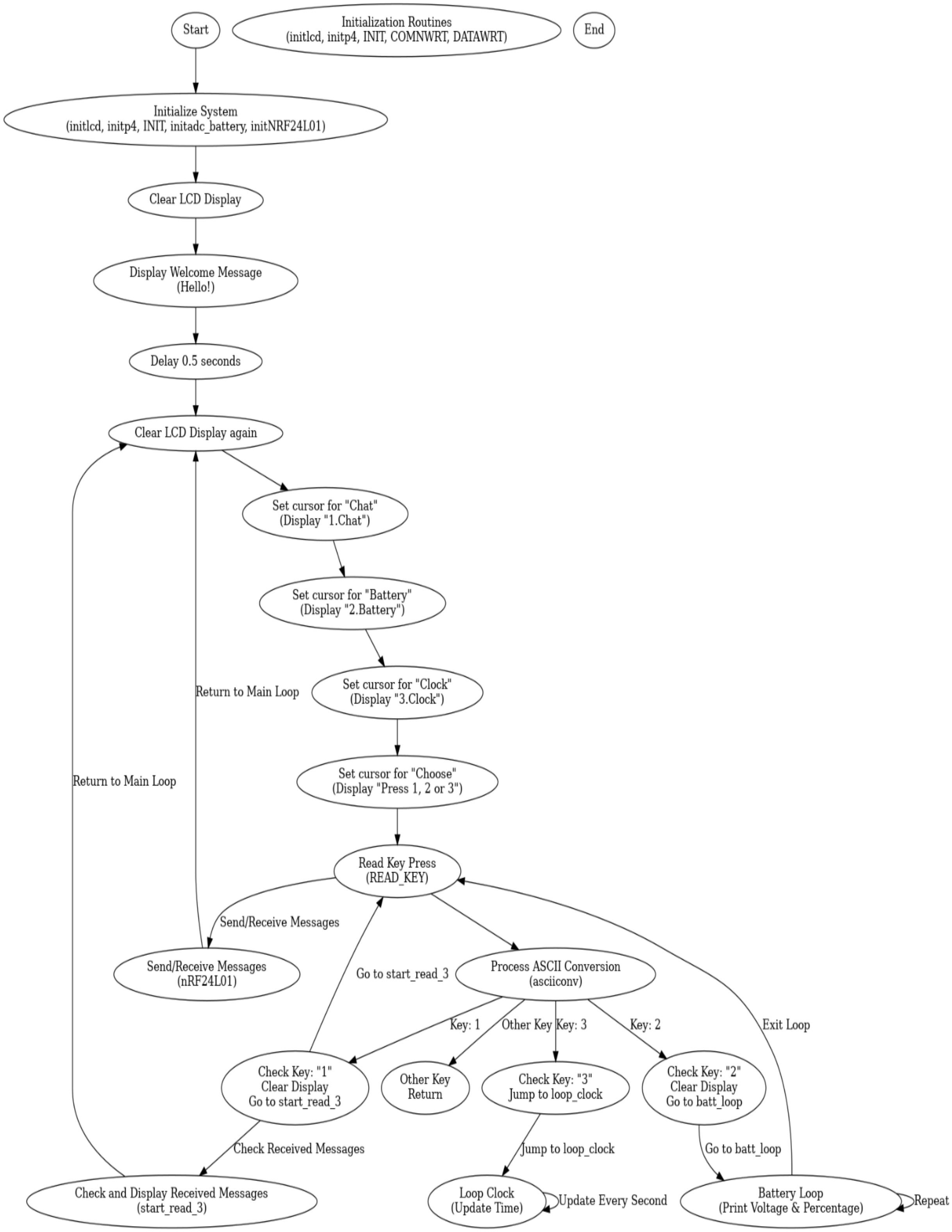
Mode	PWR_UP register	PRIM_RX register	CE input pin	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFOs. Will empty all levels in TX FIFOs[a].
TX mode	1	0	Minimum 10µs high pulse	Data in TX FIFOs.Will empty one level in TX FIFOs[b].
Standby-II	1	0	1	TX FIFO empty.
Standby-I	1	-	0	No ongoing packet transmission.
Power Down	0	-	-	-

a. If CE is held high all TX FIFOs are emptied and all necessary ACK and possible retransmits are carried out. The transmission continues as long as the TX FIFO is refilled. If the TX FIFO is empty when the CE is still high, nRF24L01+ enters standby-II mode. In this mode the transmission of a packet is started as soon as the CSN is set high after an upload (UL) of a packet to TX FIFO.

b. This operating mode pulses the CE high for at least 10µs. This allows one packet to be transmitted. This is the normal operating mode. After the packet is transmitted, the nRF24L01+ enters standby-I mode.

Table 15. nRF24L01+ main modes

- Flowchart:



Start
Initialization Routines
(initlcd, initp4, INIT, COMNWRT, DATAWRT)
End
Initialize System
(initlcd, initp4, INIT, initadc_battery, initNRF24L01)
Clear LCD Display
Display Welcome Message
(Hello!)
Delay 0.5 seconds
Clear LCD Display again
Set cursor for "Chat"
(Display "1.Chat")
Set cursor for "Battery"
(Display "2.Battery")
Set cursor for "Clock"
(Display "3.Clock")
Set cursor for "Choose"
(Display "Press 1, 2 or 3")
Return to Main Loop
Return to Main Loop
Read Key Press
(READ_KEY)
Send/Receive Messages
Send/Receive Messages
(nRF24L01)
Go to start_read_3
Process ASCII Conversion
(asciiconv)
Key: 1
Other Key
Key: 3
Key: 2
Exit Loop
Check Key: "1"
Clear Display
Go to start_read_3
Other Key
Return
Check Key: "3"
Jump to loop_clock
Check Key: "2"
Clear Display
Go to batt_loop
Check Received Messages
Jump to loop_clock
Go to batt_loop
Check and Display Received Messages
(start_read_3)
Loop Clock
(Update Time)
Update Every Second
Battery Loop
(Print Voltage & Percentage)
Repeat

- Flowchart for RF-Based Secure Wireless Communication System

Start

1. Initialize System

Initialize LCD (lcall initlcd)

Initialize Port 4 (lcall initp4)

Initialize General Settings (lcall INIT)

Initialize ADC for Battery (lcall initadc_battery)

Initialize NRF module

Clear LCD Display (mov a, #cleardisplay, lcall COMNWRT, LCALL DELAY)

2. Display Welcome Message

Set LCD cursor to a specific position (mov a, #c6h, lcall COMNWRT)

Display "Hello!" message (mov dptr, #welcome, lcall lcdoutmsga)

Delay for 0.5 seconds (mov a, #55, lcall delaya0k05s)

Clear LCD Display again (mov a, #cleardisplay, lcall COMNWRT, LCALL DELAY)

3. Main Loop

Set cursor position for "Chat" (mov a, #88h, lcall COMNWRT)

Display "1.Chat" (mov dptr, #chat, lcall lcdoutmsga)

Set cursor position for "Battery" (mov a, #c8h, lcall COMNWRT)

Display "2.Battery" (mov dptr, #batt, lcall lcdoutmsga)

Set cursor position for "Clock" (mov a, #9ch, lcall COMNWRT)

Display "3.Clock" (mov dptr, #clock, lcall lcdoutmsga)

Set cursor position for "Choose" (mov a, #d4h, lcall COMNWRT)

Display "Press 1, 2 or 3" (mov dptr, #choose, lcall lcdoutmsga)

4. Read Key Press

Call READ_KEY to read the key press

Jump to the start of the main loop

Key Press Handling (READ_KEY)

5. Set TEMPVAR to 3

Attempt to read a byte from the PS/2 keyboard 3 times

Clear the accumulator and carry flag

Wait for keyboard clock to be low then high

Read 8 bits of data into the accumulator (rotate left)

Decrement TEMPVAR and retry if necessary

Process ASCII Conversion (asciiconv)

6. Compare the ASCII value and handle specific key presses

If 'h', clear the display and return to start_read_3

If 'x', clear the display and go to batt_loop

If 'd', jump to loop_clock

Otherwise, return

7. Loop Clock (loop_clock)

Set initial time values for hours, minutes, and seconds

Clear the LCD display

Display the current time (update every second)

Convert hex time to BCD

Display hours, minutes, and seconds on LCD

Increment seconds, minutes, and hours as needed

Display "ESC to return"

Read key press to possibly exit the loop

8. Battery Loop (batt_loop)

Measure battery voltage

Print voltage and battery percentage on the LCD

Delay for a short period and read key press to possibly exit the loop

Learning Outcomes

Throughout this project, I have gained valuable hands-on experience and developed a robust understanding of several key concepts and skills in the field of embedded systems and electronics. The specific learning outcomes of this project include:

1. **Programming an LCD:**

   - Gained proficiency in interfacing and programming an LCD module to display various types of information, such as text and numeric data.

   - Learned how to send commands and data to the LCD and control its cursor and display settings.

2. **Programming a PS/2 Interface:**

   - Acquired knowledge of the PS/2 communication protocol and how to interface a PS/2 keyboard with a microcontroller.

   - Developed skills to read and interpret keypresses from the keyboard and process them appropriately.

3. **Internal Clock Management:**

   - Understood the concept of internal clocks in microcontrollers and how to configure and use them for timing and scheduling tasks.

   - Implemented clock-based routines to manage real-time applications like displaying and updating a digital clock.

4. **PCB Design and Assembly:**

   - Learned how to design a printed circuit board (PCB) using CAD software.

   - Gained experience in assembling the PCB, including placing and soldering both through-hole and surface-mount components.

5. **Soldering SMD Components:**

   - Developed proficiency in soldering surface-mount devices (SMD) using appropriate tools and techniques to ensure reliable connections.

6. **Using Interrupt Service Routines (ISR):**

   - Learned about the importance and implementation of ISRs for handling time-sensitive tasks and events in embedded systems.

- Implemented ISRs to manage tasks like checking for keypresses at regular intervals.

7. **SPI Communication:**

- Gained knowledge of the Serial Peripheral Interface (SPI) protocol and how to use it for communication between microcontrollers and peripheral devices.

- Implemented SPI communication to interface with sensors and other modules.

8. **RF Communication:**

- Understood the basics of RF communication and how to establish wireless connections between devices.

- Implemented RF modules to transmit and receive data wirelessly.

9. **Battery Measurement and Monitoring:**

- Learned how to measure and monitor battery voltage using ADC (Analog-to-Digital Conversion).

- Developed algorithms to calculate and display battery percentage based on the measured voltage.

10. **Power Management:**

- Gained insights into efficient power management techniques to optimize battery life in embedded systems.

11. **Debugging and Troubleshooting:**

- Enhanced problem-solving skills by debugging and troubleshooting various hardware and software issues encountered during the project.

12. **Documentation and Presentation:**

- Learned the importance of thorough documentation for project development and maintenance.

- Developed skills to present the project and its outcomes effectively.

This project has provided a comprehensive learning experience, combining theoretical knowledge with practical application, and has equipped me with a strong foundation in embedded systems design and development.