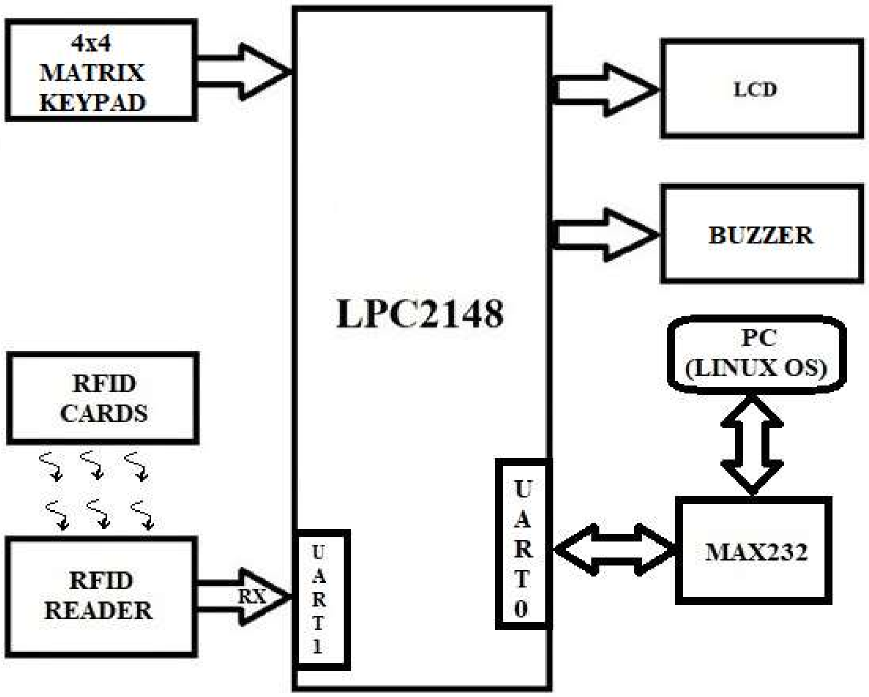
ATM SYSTEM DESIGN WITH DATABASE INTEGRATION

OBJECTIVE:

The main aim of this project is to develop a **secure ATM system** using **RFID authentication** and a **PIN-based interface**, with backend **banking database integration implemented in C using data structures** (instead of SQL-based systems).

BLOCK DIAGRAM:



REQUIREMENTS:

HARDWARE REQUIREMENTS:

- LPC2148 Microcontroller
- RFID Reader
- RFID Cards
- 16x2 LCD Display
- 4x4 Matrix Keypad
- MAX232 (for UART level shifting)
- USB-to-UART Converter/DB-9 Cable
- Buzzer

SOFTWARE REQUIREMENTS:

- EMBEDDED C – Programming
- KEIL uVision IDE
- FLASH MAGIC
- GCC Compiler

STEPS TO BE FOLLOWED TO COMPLETE THE PROJECT:

1. Create a new folder on your PC and name it with your project title.
2. This project includes two application programs:
   - One for the microcontroller board (ATM front-end)
   - One for the PC side (Linux), written in C, to simulate a banking database using data structures and file handling.
3. Copy and verify the following hardware interface modules into your MCU project folder:
   - lcd.c, lcd.h, delay.c, delay.h, uart.c, uart.h, keypad.c, keypad.h
4. Individually test all modules:
   - LCD: Display characters and strings.
   - Keypad: Read and display input values.
   - UART: Send/receive test strings using UART0 and UART1 (via interrupt).
   - RFID: Read card data via UART1 and display it on LCD.
5. Finalize your main microcontroller code projectmain.c:
   - Display project title briefly.
   - Continuously wait for RFID card to be presented.
   - When card is read, send the card number to PC over UART in the mentioned format. (#CARDNUMBER$)
   - Wait for PC to validate and reply.
   - If PC responds with success, display "Enter PIN" on LCD.
   - Read PIN from keypad.
   - Send card number and PIN to PC over UART in the format:

     #CARDNUMBER#PIN$
   - Wait for PC to validate and reply.

6. If PC responds with success, show the user menu on LCD:

    1. BALANCE
    2. DEPOSIT
    3. WITHDRAW
    4. EXIT

    - User selects an option via keypad.
    - Send request in format: #ACTION#AMOUNT$

7. Wait for PC to respond with success or error message.
    - Display response on LCD.
    - Loop back to initial state (waiting for new card).

PC-SIDE PROGRAM IN C (DATABASE USING DS):

1. Write a C program on the PC to:
    - ➢ Receive UART data from MCU.
    - ➢ Parse RFID and PIN values.
    - ➢ Match against an array of User structures.
    - ➢ Validate and update balance or return failure.
2. Use linked list to maintain transaction logs (deposit/withdrawals).
3. Use file handling (users.txt, transactions.txt) to store account and log data persistently.

Example Structures:

```
typedef struct {
   char tag_id[20];
   char pin[10];
   char name[50];
   float balance;
} User;


typedef struct Transaction {
   char tag_id[20];
```

```
    char type[10]; // "Withdraw"/"Deposit"
    float amount;
    char timestamp[25];
    struct Transaction *next;
} Transaction;
```

4. Send the result of operations back to the MCU in format:
   - @OK#BALANCE:1234.56$
   - @ERR#INVALID_PIN$
   - @ERR#INSUFFICIENT_BALANCE$

MESSAGE FORMATS:

From MCU to PC:

#CARD:12345678#PIN:4321$

#TXN:DEPOSIT#AMOUNT:1000$

From PC to MCU:

@OK#BALANCE:5300$

@ERR#INVALID_PIN$

@ERR#INSUFFICIENT_BALANCE$

This process provides a brief and structured outline of the project implementation. Based on specific application requirements, additional functionalities and enhancements may be incorporated into both the microcontroller and PC-side applications.

************************ **ALL THE BEST** *************************