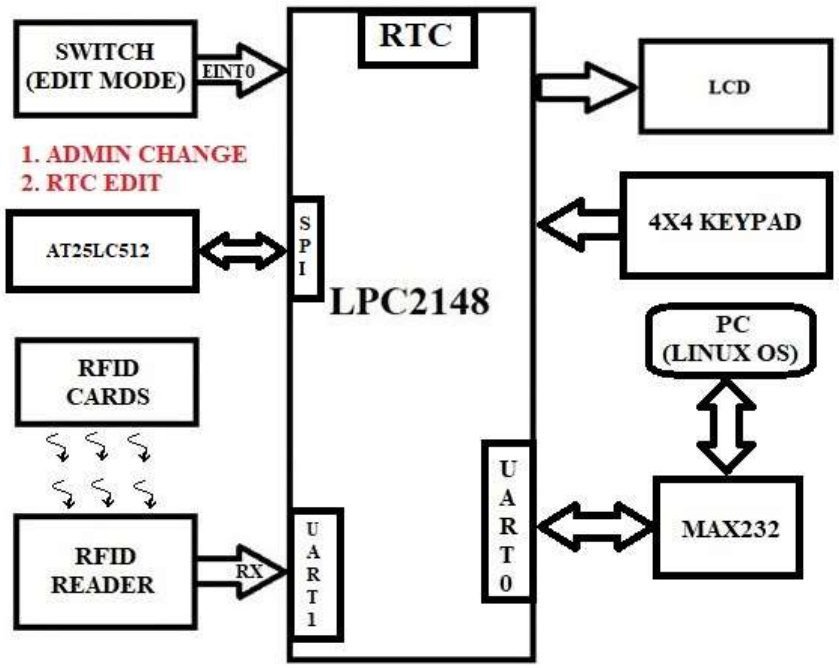
RFID-BASED ATTENDANCE SYSTEM WITH DATABASE INTEGRATION

AIM:

The main aim of the project is to develop the general attendance system by using RFID technology with data base integration.

BLOCK DIAGRAM:



REQUIREMENTS:

HARDWRAE REQUIREMENTS:

- LPC2148
- RFID READER
- RFID CARDS
- LCD
- SWITCHES
- 4X4 KEYPAD
- MAX232
- AT25LC512
- USB-TO-UART CONVERTER

SOFTWARE REQUIREMENTS:

- EMBEDDED C – PROGRAMMING
- KEIL-C COMPILER
- FLASH MAGIC

Steps to be followed to complete your project:

- Create New Folder in your server save that folder with your project name.
- In this project two different application programs are required to communicate to get the final result. One application program is for microcontroller board and other one C program which is used for communicating the microcontroller board with UART communication and database integration from LINUX OS.
- Follow the below mentioned steps to complete the project.
- Copy what you done files like lcd.c, lcd.h, delay,c, delay.h, uart.c, uart.h, keypad.c, keypad.h, spi.c, spi.h, spi_eeprom.c and spi_eeprom.h into project folder.
- Individually can check each and every module.
- First check lcd to display character constant, string constant and integer constant.
- Next check keypad peripheral by displaying key values on LCD.
- Next check the SPI based EEPROM working condition by storing one user defined string in to EEPROM fixed locations and same data read back and display it on LCD. If display message is belonging to written message, then EPPROM working is fine.
- Next check uart0 and uart1 peripherals by transmitting character constant, string constant and receive string constant using hyper terminal. (Note: use UART interrupt concept for both the UART's (UART0 & UART1))

  Note: By using UART0 driver code, you need to develop UART1 driver.
- Connect RFID READER D0 pin to USB to UART converter and test the working condition.
- Then write logic for reading the card number using UART1 interrupt and display it on LCD.
- Then check the external interrupt (EINT0) working condition on hardware.
- If above steps are completed create new file with projectmain.c, add all peripheral definition files, and write below steps in projectmain.c file
- Initially project name will display on LCD for some specific time and read the ADMIN card number from the EEPROM and store it in one variable. In continuous process, application program is waiting for card (ADMIN card or USER card) from the user. If ADMIN card placed, then this card number will send to the PC through UART0 in specific format (**A**CARDNUMBER**$**). In Linux system, once it receives the ADMIN card then application program is giving support for updating the new user information into the database based on user input. To know more details about Linux code please refer the below LINUC CODE working process. If USER card is placed, then this card number will send to the PC through UART0 in specific format along with RTC information (**U**CARDNUMBERRTCINFO**$**).

And it is waiting for the reply from the PC. If it receives the reply (positive/negative), it will display on LCD then again waiting for card (ADMIN/USER). Due to any reason, if it required to change the ADMIN card number, then need to generate the interrupt by pressing switch which is connected to the interrupt pin. Whenever interrupt raised, then below mentioned menu will display.

MENU:

1. ADMIN CHANGE
2. RTC INFO CHANGE

Based on the user requirement, need to select any one of the options from the keypad. If option 1 is selected, then application program is waiting for card. Here need to give the new ADMIN card entry. After card is placed, that card number is updated as new ADMIN card and card number is saved in the EEPROM locations where old ADMIN card number is saved. If user is selected the option2, then edit the required RTC information from the keypad. (Note: required to add all testcases for this editing process).

- LINUX CODE WORKING: First download the UART_LOOPBACK_CODE_FOR_LINUX code from LMS card under reference programs and check the working of that code on Linux system.
- Initially create one .csv file. And update the required fields in that file. After executing the Linux code executable file, initially it will display the all-users information on terminal and waiting for the card information from the microcontroller board. Based on the card it received, it will proceed accordingly. If received card is belongs to ADMIN card, then display the below mentioned menu.

MENU:

1. ADD USER
2. DELETE USER
3. EDIT USER
4. EXIT

All the above-mentioned options are required to complete manually. If option 1 is selected, need to take the new user details (user ID (RFID CARD number), USER Name) and update it in to .csv file. By mistake any wrong updates were done then use option3 (EDIT USER) for the modifications/editing. Option 2 is used for deleting the exited user. If option 4 is selected then send the admin exit related message to controller board.

If user card received, then update user IN/OUT timings in to .csv file to the respective user equivalent columns. Refer the below mentioned details for .csv file information.

S.No	USER ID	USER NAME	DATE	WORKING HOURS	IN/OUT STATUS	IN	OUT	IN	OUT

Initially, IN/OUT status is 0 (OUT). If any user card received, the read the IN/OUT field status and update the new status in that field then update the date and IN/OUT time in to respected fields.

For every IN&OUT need to calculate total time and update it in the WORKING HOURS field.

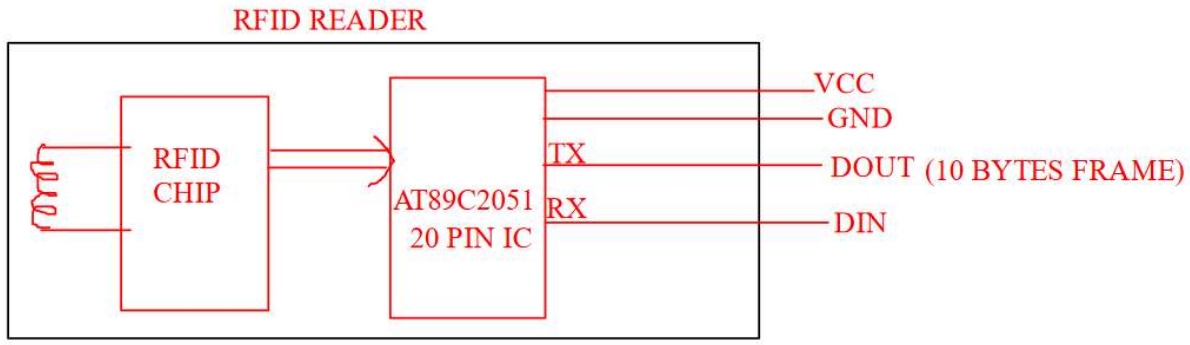
For every card it received from the microcontroller board, need to do the required operations and finally need to send the acknowledge message to the controller board. Better to use one specific syntax for the reply message so that it is giving benefit for the controller board programming.

- If you're getting this output then your project is completed.

Note: Provide some dummy user IDs in-between the USER ID'S so that it is giving clear picture about execution time related to different USERS. Then think about how to reduce the execution time through your application program.

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

RFID READER: RFID Reader will send the 10 bytes of data once card is placed nearer to the RFID Reader. Once check the below images to get some idea on RFID Reader block diagram. Refer the supported data sheets to get some more knowledge.



For example, card number is 12345678, then output of the RFID Reader is **0x02 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x03** (hex format)