GüçlüKadro

CENG 491

SENIOR DESIGN PROJECT

FINAL DESIGN REPORT

GROUP MEMBERS

RECEP ALABAŞ

YAŞAT KADİR BİLGİN

BERK ÇAKIR

ÖZER DURMAZ

INDEX

1) INTRODUCTION

1.1) PURPOSE OF THIS DOCUMENT

This document is prepared to supply a final design to our project.

The purpose of this document is to show our design process and the solutions for the design problems.

1.2) PROBLEM REDEFINITION

Wireless communication is a rapidly growing part of the communication systems. While the wireless communication quality improves, the demand of using wireless technology increases. For that reason wireless access points have an important part at the market. However, it is important to develop a different type of wireless access point to have a big portion from the market. Our prototype will be different by its LCD monitor. The user will able to access some kind data from this monitor namely, the number of users, their names and their connection speeds connected to the access point. Also, it will be able to give information about their connection period of times and about the errors in the connection. Moreover, the user will be able to block some unwanted Mac addresses.

1.3) DESIGN GOALS AND OBJECTIVES

a) Extensibility:

However we showed our designs in this report, they are not very strict ones. Our designs have extensibility in order to add new specialties to the project. We tried to design our prototype considering that any other improvement will be added later.

b) Robustness:

The product should be able to deal with invalid user inputs. To illustrate, our product will be able to manage with pushing of its buttons at the same time, in other words wanting different data at the same time. It will be able to do error checking and returns this error reports to the user by the LCD display.

c) Reliability:

We are planning to program our product, as reliable as possible. In order to accomplish this aim, we will test our program many times. We think that we will succeed to produce a bug free product.

d) Functionality:

Our product's functions were mentioned in the Requirement Analysis Report. Our product will function in the manner that was visited in that report.

e) Usability:

Usability is one of the most important issues in our project. Our general user interface should affect the user by its easiness. The user will not encounter with any complexity when using the product. The user will sense the familiarity and the user-friendliness of the menus.

1.4) DESIGN CONSTRAINTS

a) Performance:

Due to the fact that performance is a very important issue for users, we considered the efficiency of our product in our designs. Our product will work very efficient and show high performance.

b) Time:

According to the Ceng 491 syllabus, our project must be finished by June. Because of that reason we must work hard and provide a prototype at the end of this semester. We must follow our plans stated in our Gantt chart in this report.

c) Funding:

Computer Engineering Department is going to pay us about 50$ for our project expenditures. Until now, we bought a 128x64 graphic LCD, RS-232 to USB converter and components of the printed circuit board. These didn't cost as much as the fund, so we are able buy our other necessities.

d) Resource:

We need some software and hardware resources in order to complete our project. Software resources are supplied by our teaching assistant and the company. Also, we can provide the hardware resources by the help of our teaching assistant. In addition to them, we need some books and articles to learn the details of our working scope. Every member of the project including the company, and the teaching assistant are providing us these kinds of useful documents. Also, we are trying to find documents via internet and Metu Library.

2) HARDWARE DESIGN

The hardware design of our project consists of two main parts (Figure 2.1). First one is Airties AP400 access point. This is the access point that we are going to use it with special hardware in order to design a model of an improved access point. The hardware of AP400 is discussed in the next section. The second part is the special hardware we are going to use to construct the model. It will consist of a PCB (printed circuit board) and the required hardware (a microcontroller, an LCD, an oscillator, etc.) on it. The details of the PCB are discussed in the next chapter. AP400 will be connected to the PCB with an RS232 cable. With this connection the needed information about the network traffic will be transferred to the board and will be displayed on the graphic LCD. Also there will be some buttons on the board which will be used to change the screen, to pass from one information display to another (e.g. while the default information, date and time is displaying, pressing the button will show the users connected at that time.) The type of the microcontroller we are going to use is pic16f877, just because it can fulfill our demands and we are familiar it from ceng336.

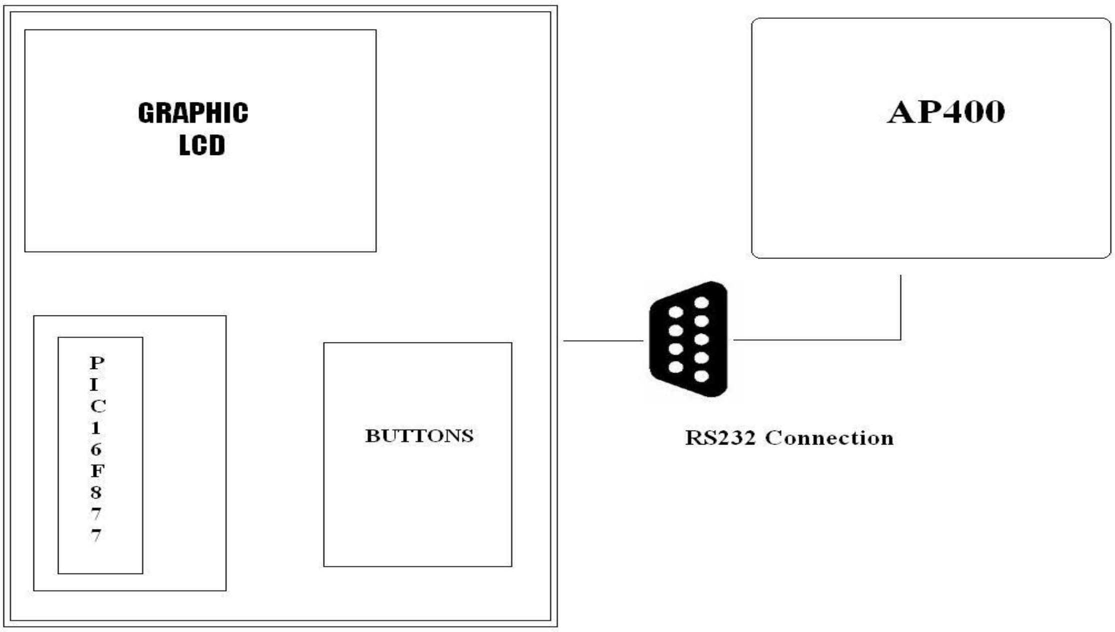


Figure 2.1

2.1) REQUIRED HARDWARE

2.1.1) AIRTIES AP400

Airties AP400 is an access point (a special hardware software combination which can connect wireless communication devices) that is an embedded system design running Linux. It is IEEE 802.11b/g compliant. It has WPA (Wi-Fi Protected Access), WPA2 (IEEE 802.11i), WEP (Wired Equivalent Privacy) encryption and 802.1x standards. Also AP400 has ADM6996f, four-port 10/100 Mbps TX/FX plus two 10/100 MAC port Ethernet switch controller with all ports supporting 10/100 Mbps Full/Half duplex, and this chip has its own embedded memory.

Picture 2.1

2.1.2) MICROCONTROLLER

a) PIC16F877 Pin Diagram

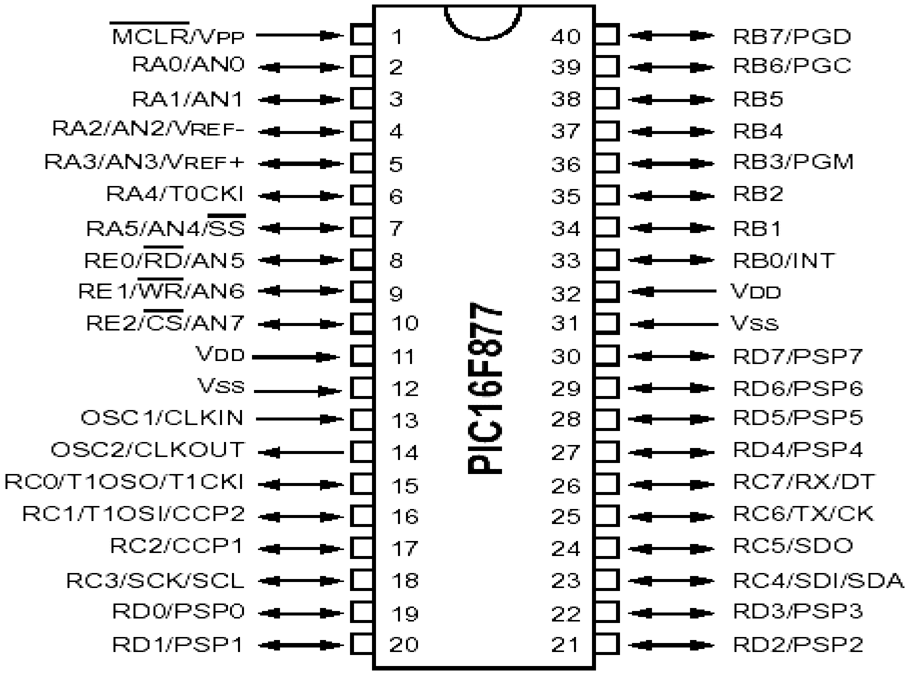


Figure 2.2

b) PIC16F877 Main Features

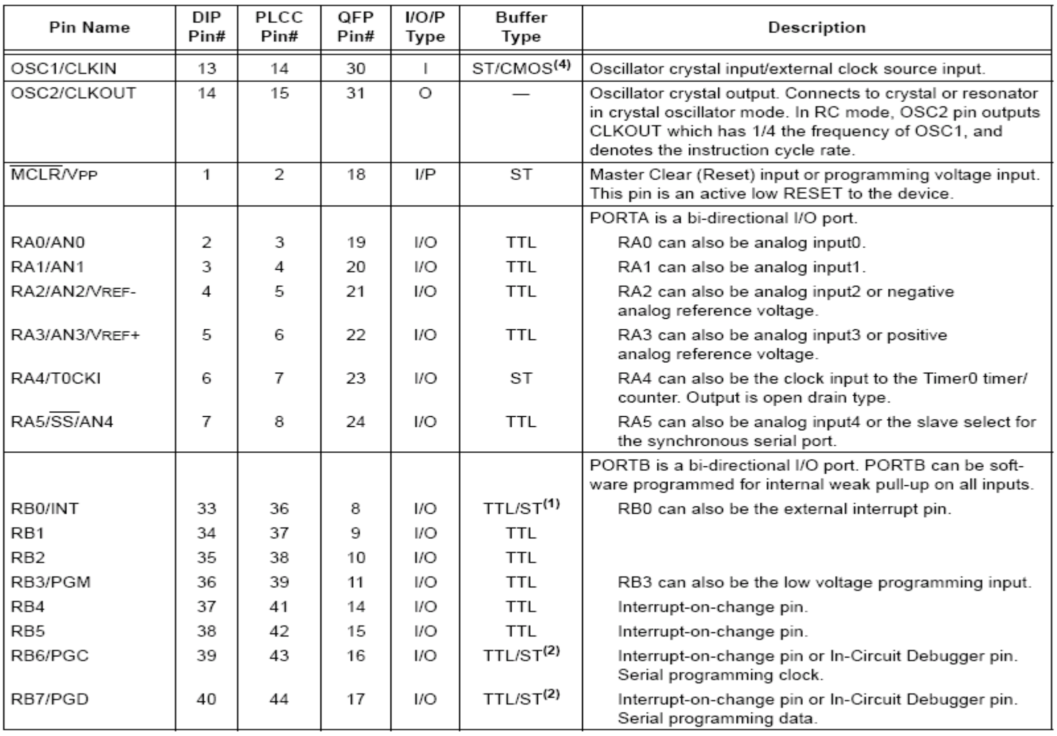
- Operating speed: DC - 20 MHz clock input

  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,

  Up to 368 x 8 bytes of Data Memory (RAM)

  Up to 256 x 8 bytes of EEPROM Data Memory
- Interrupt capability (up to 14 sources)
- Wide operating voltage range: 2.0V to 5.5V

c) Pinout Description



Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS$^{(4)}$	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/ counter. Output is open drain type.
RA5/$\overline{\text{SS}}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST$^{(1)}$	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST$^{(2)}$	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST$^{(2)}$	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

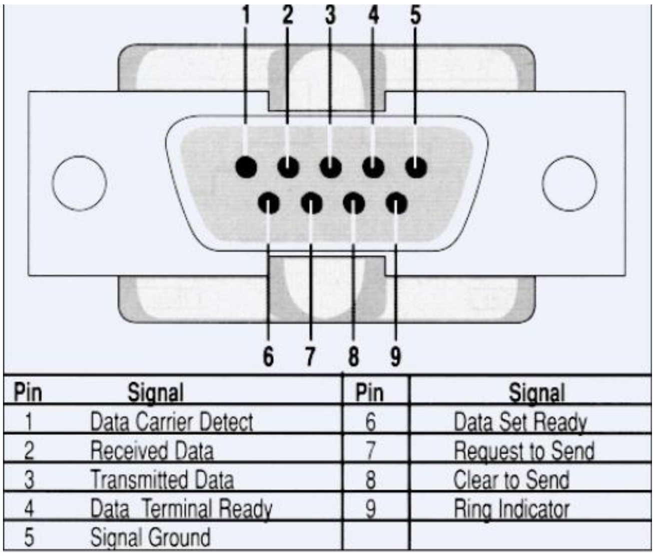
Figure 2.3

d) PIC16F877 Functionality in Our Design

The microcontroller on our PCB will be responsible for three major operations. The communication with AP400 (data transfer between AP400 and the board), ensuring the LCD to display the information get from AP400 (the suitable graphic display about the network traffic), and to change the screen when a button is pressed (this will be done by interrupts).

e) RS232

The communication between our designed board and AP400 will be established in RS232 standards. In RS-232, data is sent as a time-series of bits. Both synchronous and asynchronous transmissions are supported by the standard. The connector type we are going to use is a DE-9 (D-subminiature 9 pin**)** connector and its pinout is described in Figure 2.4. Since the voltages of the board and AP400 are not the same (5V-3.3 V) the voltage translation will be done by MAX3378 integrated circuit.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 2.4

2.1.3) GRAPHIC LCD

The network information will be displayed on the LCD. The LCD we are going to use is HY-12864K-202-R model 128x64 graphic LCD (Figure 2.5).

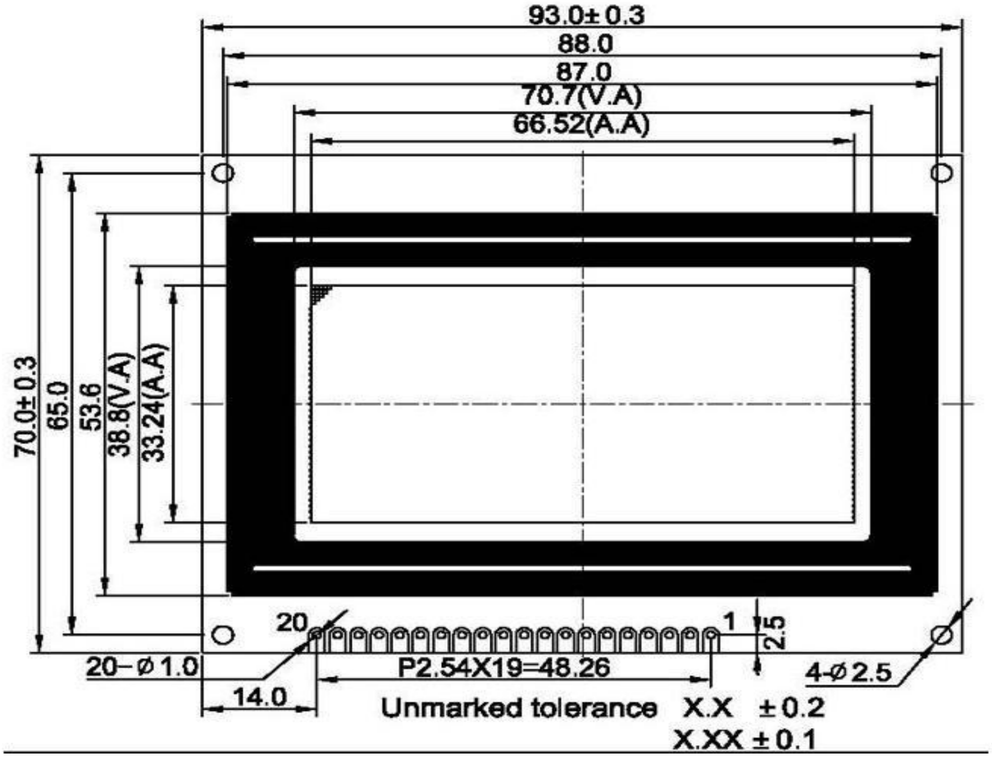
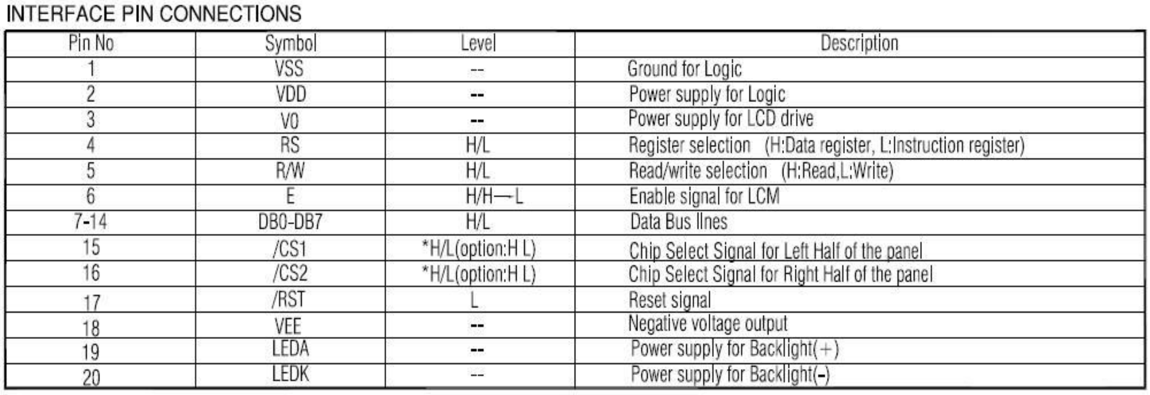


Figure 2.5



INTERFACE PIN CONNECTIONS

Pin No	Symbol	Level	Description
1	VSS	--	Ground for Logic
2	VDD	--	Power supply for Logic
3	V0	--	Power supply for LCD drive
4	RS	H/L	Register selection (H:Data register, L:Instruction register)
5	R/W	H/L	Read/write selection (H:Read,L:Write)
6	E	H/H—L	Enable signal for LCM
7-14	DB0-DB7	H/L	Data Bus lInes
15	/CS1	*H/L(option:H L)	Chip Select Signal for Left Half of the panel
16	/CS2	*H/L(option:H L)	Chip Select Signal for Right Half of the panel
17	/RST	L	Reset signal
18	VEE	--	Negative voltage output
19	LEDA	--	Power supply for Backlight(+)
20	LEDK	--	Power supply for Backlight(-)

Figure 2.6

This LCD has 20 pins and the pinout description is displayed above (Figure 2.6).

Besides these components, there will be some other needed components too, on the board like an oscillator, power jack, reset button etc. but these are so simple components and not worth to discuss with details.

2.1.4) PCB DESIGN SOFTWARE

To build a PCB, first we must design and test it with software. Our research about PCB design software is finished and we have chosen one to use. Considering that it is world's first unified electronic product development system that allows engineers to take a design from concept to completion within a single application, Altium designer will be our design software.

Main features of Altium Designer:

- ✓ Synchronous design at PCB and FPGA (Field-Programmable Gate Array, a type of logic chip that can be programmed) levels, from concept to design in one application.
- ✓ With its functional user interface, hierarchical, multi-channel schematic capture.
- ✓ Fast PCB design with Pre- and post-layout signal integrity analysis after adding components.
- ✓ Situs Topological auto routing
- ✓ A design library including over 68,000 IP components and versatile archive for schematic level FPGA design.
- ✓ Versatile project management, design tools with bidirectional synchronization from schematic to PCB and from PCB to FPGA.
- ✓ Automatic and interactive FPGA pin optimization for routing.
- ✓ PCB-FPGA I/O synchronization.

In order to have the PCB, firstly we must create a SchDoc like in figure 2.7. This figure shows the overlay of our design.

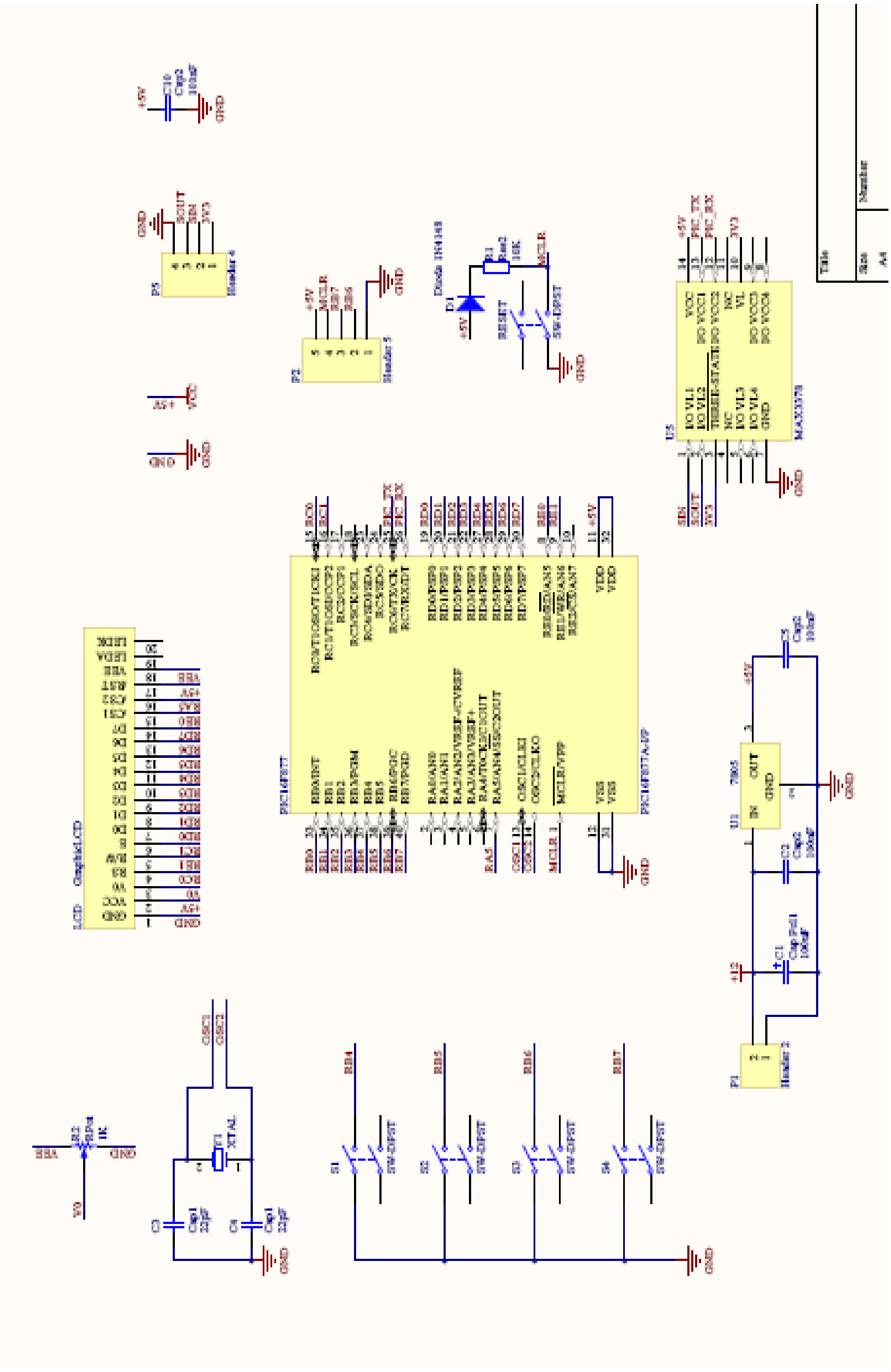


Figure 2.7

After constructing the SchDoc, we are going to design our PcbDoc which will be like in figure 2.8. In this figure, the position of the components and the nets will be finalized. As seen from the figure almost all the components will be under LCD display in order to have a simple look.

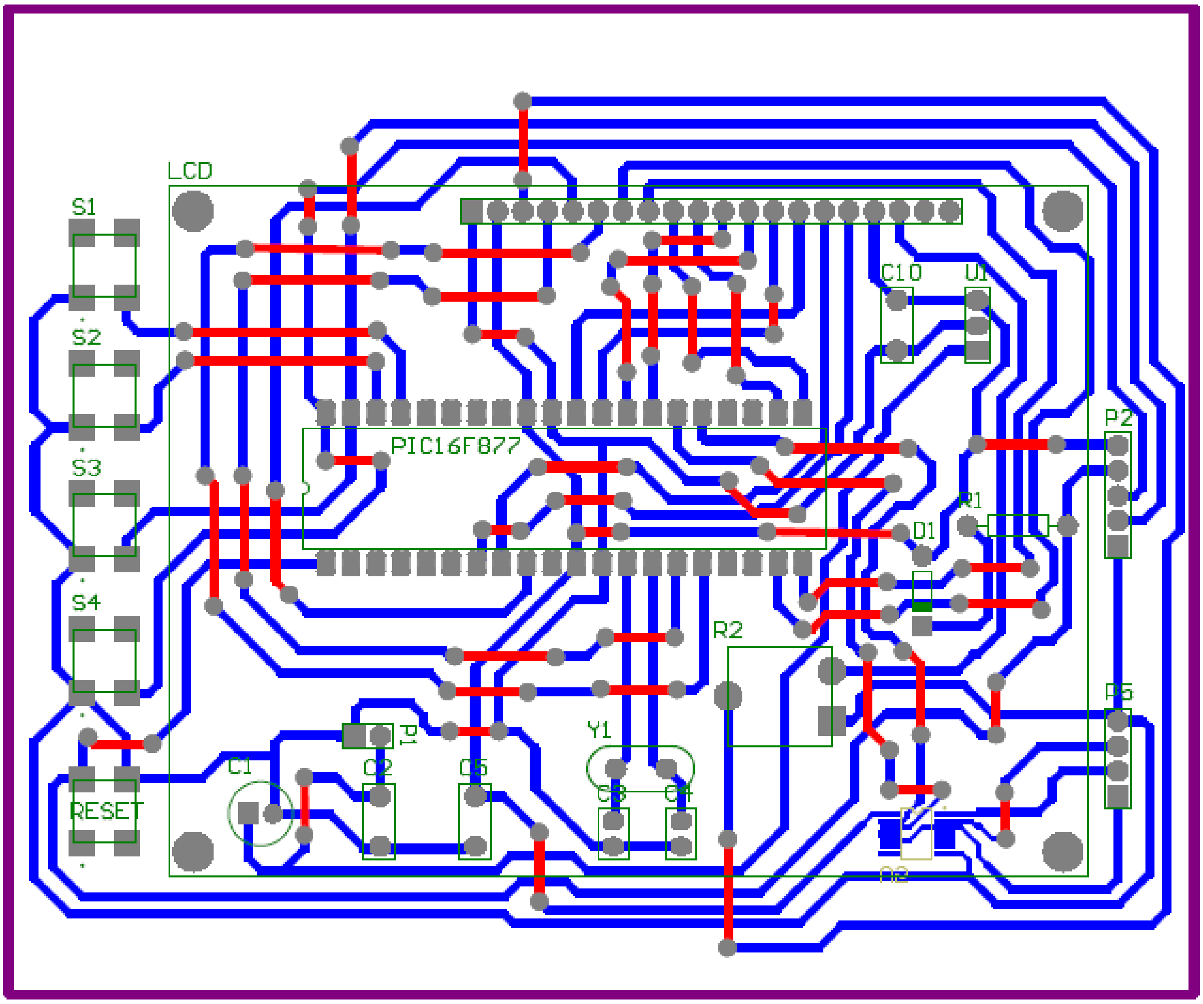


Figure 2.8

3) SOFTWARE DESIGN

Since AP400 has an embedded Linux system in it, all our programming part will be done in Linux environment. AP400 runs Linux and it has its own root file system (the necessary files needed to for the Linux system to run). There are some directories for some purposes (/bin includes the essential binaries considered part of the system, whereas /etc includes system configuration files.) Since our scripts will be put in one of these directories, their properties are very important for the system run properly. (E.g. if the script is put in /tmp directory, it will be deleted after AP400 is restarted.) As all other projects, the details are important and the details of our software design are discussed in the later sections.

3.1) REDBOOT

Redboot is an open source application for embedded systems. AP400 uses Redboot. It is based on the eCos Hardware Abstraction Layer. In our design, it is important because Redboot allows loading and execution of a kernel or executable from either tftp or via the serial console. In order to reload the root files to AP400, we use Redboot. Our final product will include the existing files of the kernel plus our scripts. The user will reload the kernel by Redboot and it will be ready to use.

3.2) PROGRAMMING ISSUES

Since our project is building a model of an improved AP400, we will add new functions to it in order to aim more customers. This “adding new functions” part will be done by our scripts. AP400 runs Linux and it has its own scripts (some of which can be useful for us) and we will add new scripts next to them (transferring this scripts from computer to AP400 is discussed in “Special Software for Software Design” section). These scripts will get the needed information about the network traffic when it is wanted (the related button on the PCB is pressed). Since many of our goals need to keep the related information since the time AP400 was opened, also we will keep this information, by a special algorithm in order to minimize the size of it, in AP400. As all other embedded systems, memory management is an important criterion for us, because the capacity of AP400 for running the scripts and keeping the information is relatively small for a normal computer program. So we will be careful about our scripts and use some algorithms to minimize the memory space we need to run the scripts and keep the information.

The programming language we are going to use may change during the project. But we believe that two of them can fulfill our demand. We will use C Programming and Bash Scripting. When we use C for the programs, we will compile it with MIPS gcc cross compiler and put its executable into one of the directories. On the other hand, Bash scripting is relatively faster than C; we will just put the script into the directory we want.

A Simple Pseudo-Code for Out Main Script

```
mainscript:
        if IdleTime is bigger than IdleTimeConstant
                DisplayDateandTime
        else
                WaitforInterrupt
```

Here *WaitforInterrupt* is responsible for the interrupts. As mentioned in the hardware design chapter, interrupts will be used when user wanted one of the information about the network traffic to be displayed. This will be done by buttons on the PCB. According to the type of the button *WaitforInterrupt* will call the suitable script to get the needed information. *DisplayDateandTime* function is for the displaying the current date and time, when the user did not press any buttons for some time.

3.2.1) HOW TO GET THE INFORMATION

As mentioned above AP400 runs Linux and has its own scripts. Some of these scripts, for example, are the result of a project, The Linux Wireless Extension and the Wireless Tools, an open source project. These project codes may be example of writing wireless tools with C. We may write some programs like these ones to get the information, compile it, and transfer it to the AP400; or we can use the outputs of these tools with a suitable script (that we write with Bash Scripting). These are two examples of getting the network information. For example wlanconfig command returns the user mac addresses, with some other information and display them on the screen. Since we need to display only the mac addresses of the users, we can do this in two ways; first we can rewrite the C code of wlanconfig in order to get only mac addresses, or second we can get the output of wlanconfig with a Bash script and choose only the mac addresses.

3.2.2) PIC PROGRAMMING

After getting the information and transferring it to the PCB, a program is needed to display the information. We are going to use C for this program and it will be suitable with our microcontroller, PIC16F877.

3.2.3) SPECIAL SOFTWARE FOR SOFTWARE DESIGN

For the communication between the computer and AP400, we use a graphical serial terminal program, cutecom. The file transfer, e.g. transferring a script from the computer to AP400 or reloading the root files to the AP400 can be done by entering the commands from cutecom and by help of a tftp server (for a fast transfer). The communication between the computer and AP400 is established with an RS232 cable.

Some features of cutecom:

- Ctrl+C, Ctrl+Q and Ctrl+S control sequences work,
- input history,
- xmodem, ymodem, zmodem support (requires the sz tools)
- hexadecimal input and output

In order to compile the pic program we are going to use SDCC, a retargettable, optimizing ANSI - C compiler. SDCC supports PIC16 and PIC18 series. It is free open source software.

To load the .hex file to the pic, we are going to use winpic800. We will use it because we are familiar to it from ceng336 and it is easy to use.

3.3) MEMORY ISSUES:

Since we are working on embedded Linux, we must be more careful about memory problems, because we have relatively small memory to keep our scripts, and network data. We have been told that we have 50KB free space for these in the access point, which is enough for the scripts but we will not be able to hold large amounts of connection data, like the usage history of the device. So we will display this information for a certain period of time, and will clear the history at the end of the period.

4) MODELING

The related network traffic, network properties, and user data will be kept in AP400. The data diagram is above. User has his name, mac address, IP, and status of the connection. One user can connect exactly one network, but the network can accept zero or more user connections. The traffic data consists of last day upload, last day download, and data transmit rate. Exactly one user can change the network properties and traffic data information. Below diagram shows the data relationships.

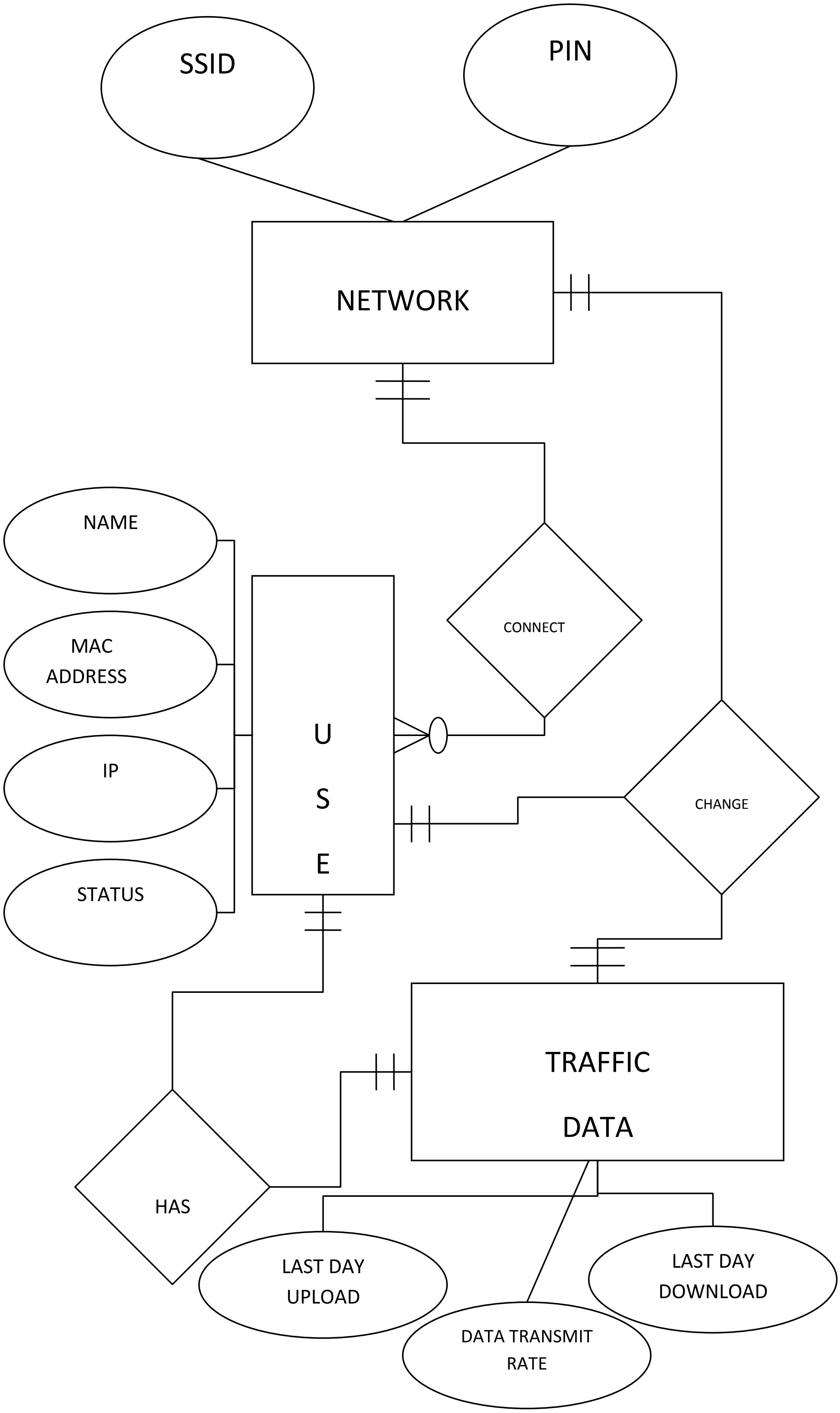

SSID
PIN
NETWORK
NAME
MAC ADDRESS
IP
STATUS
U S E
CONNECT
CHANGE
TRAFFIC DATA
HAS
LAST DAY UPLOAD
DATA TRANSMIT RATE
LAST DAY DOWNLOAD

Figure 4.1

4.1) USE CASE DIAGRAMS

4.1.1) USE CASE DIGRAM I

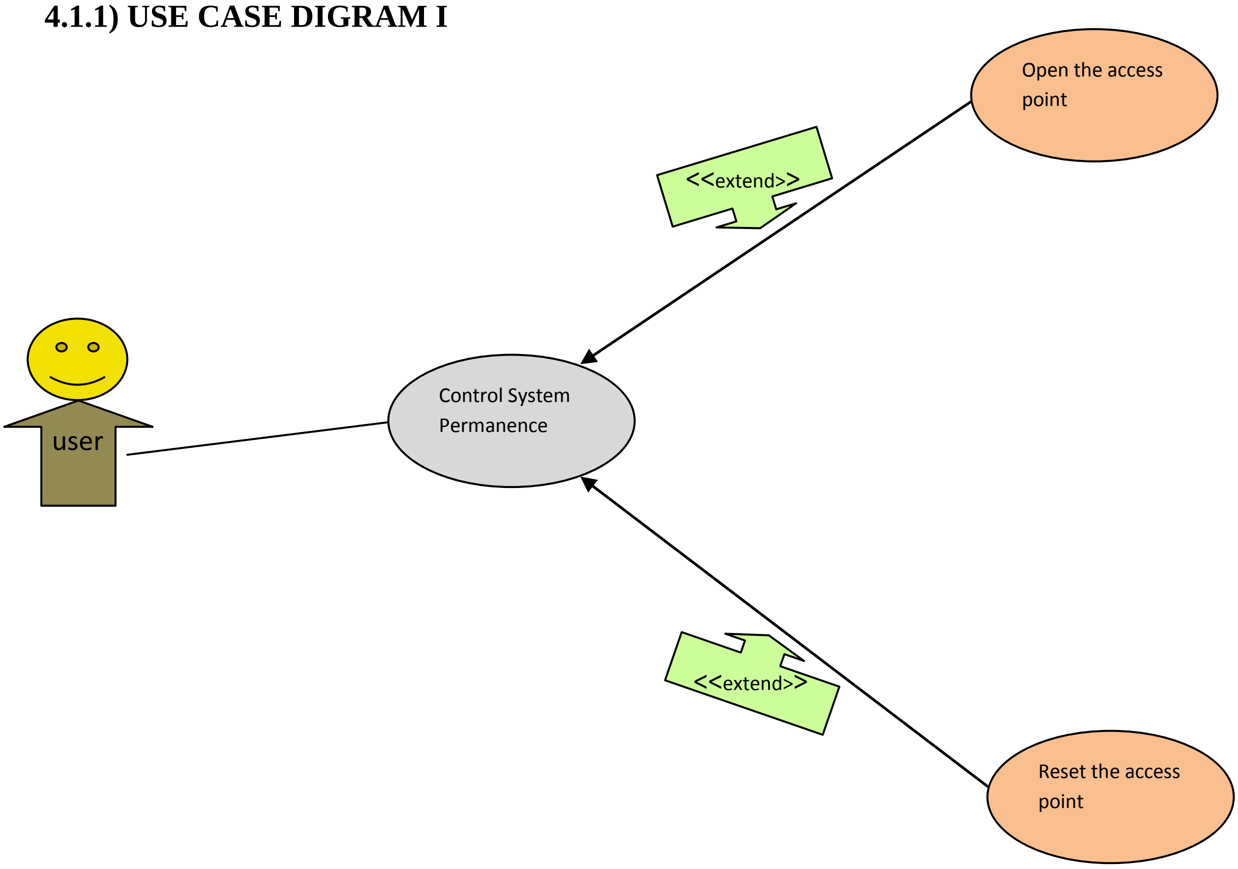


Figure 4.2

Managing system is one of the vital points of a project. In our project user must also act as admin of the system. He or she must be responsible for the permanence of the system. To make the system available every time user must take some action. These actions can be listed as followed:

- **Open the access point**: User must open access point before clients connect to it. This may be seen as very easy role but it is important to initialize access point properly.
- **Reset the access point:** To avoid the clustering of data user must reset the system periodically. Also this action will raise the system performance.

4.1.2) USE CASE DIAGRAM II

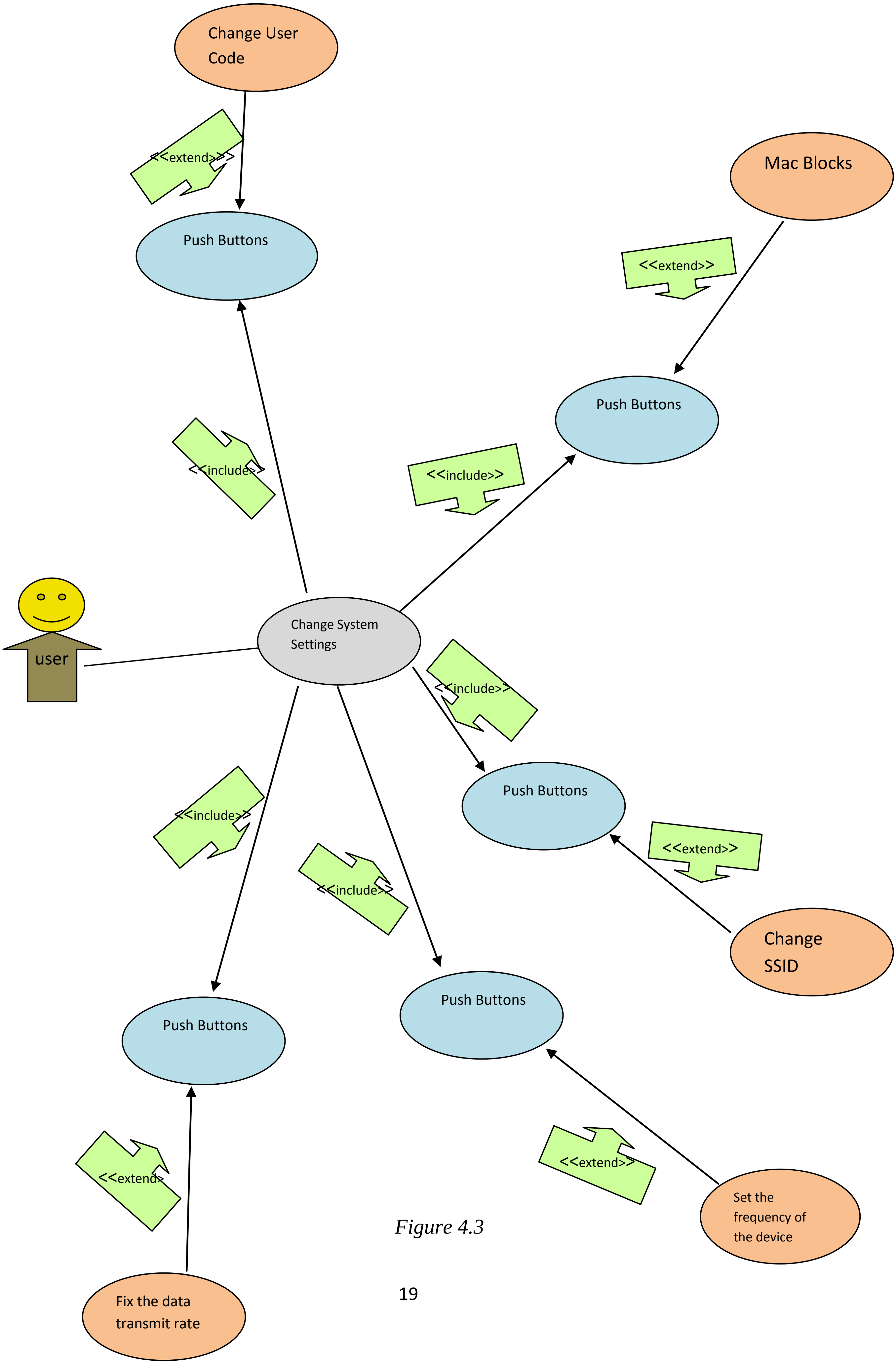

Change User Code
<<extend>>
Push Buttons
Mac Blocks
<<extend>>
Push Buttons
<<include>>
<<include>>
user
Change System Settings
<<include>>
Push Buttons
<<extend>>
Change SSID
<<include>>
<<include>>
Push Buttons
Push Buttons
<<extend>>
<<extend>>
Set the frequency of the device
Fix the data transmit rate

Figure 4.3

Another important point of our project is changing system settings. User must configure the system for maximal performance and security, because every action is performed according to these settings. In our project user can change the pin and SSID of the system, manage the connections, set frequency of device, fix the data transmit rate. Also if user wants he or she can block some Macs. All these actions are very beneficial to the system security and performance.

- **Change User Code:** User can change the pin to increase the security level of the system. Because a stable pin can be easily stolen. Changing pin regularly will block the unauthorized user to connect the system.
- **Manage Connection:** User can manage the connections to avoid over-using of the system. When user notices that some users make a lot of jam in the network he or she can warn the user or do something to avoid this. At this point our system will only show other connected user’s connection status to the user. Then, immediately he or she can take precautions to overcome this problem.
- **Mac Block:** If a connected user makes unwanted action then it will be possible to block his or her Mac. Disturbing other connected users, keeping network busy can be examples of unwanted action.
- **Change SSID:** User can change the SSID of the system to make network more user-friendly. For example a company can change the SSID of the system to their company name.
- **Set Frequency and Channel Of The Device:** Which channel to use and its frequency can be set by user. User can coordinate network traffic by this action. It will be very beneficial to maximize the network usage.
- **Fix The Data Transmit Rate:** User can fix the data transmit rate in order to have a single transmit rate. For example, the entire rate other than the user specified will automatically become equal to it.

4.1.3) USE CASE DIGRAM III

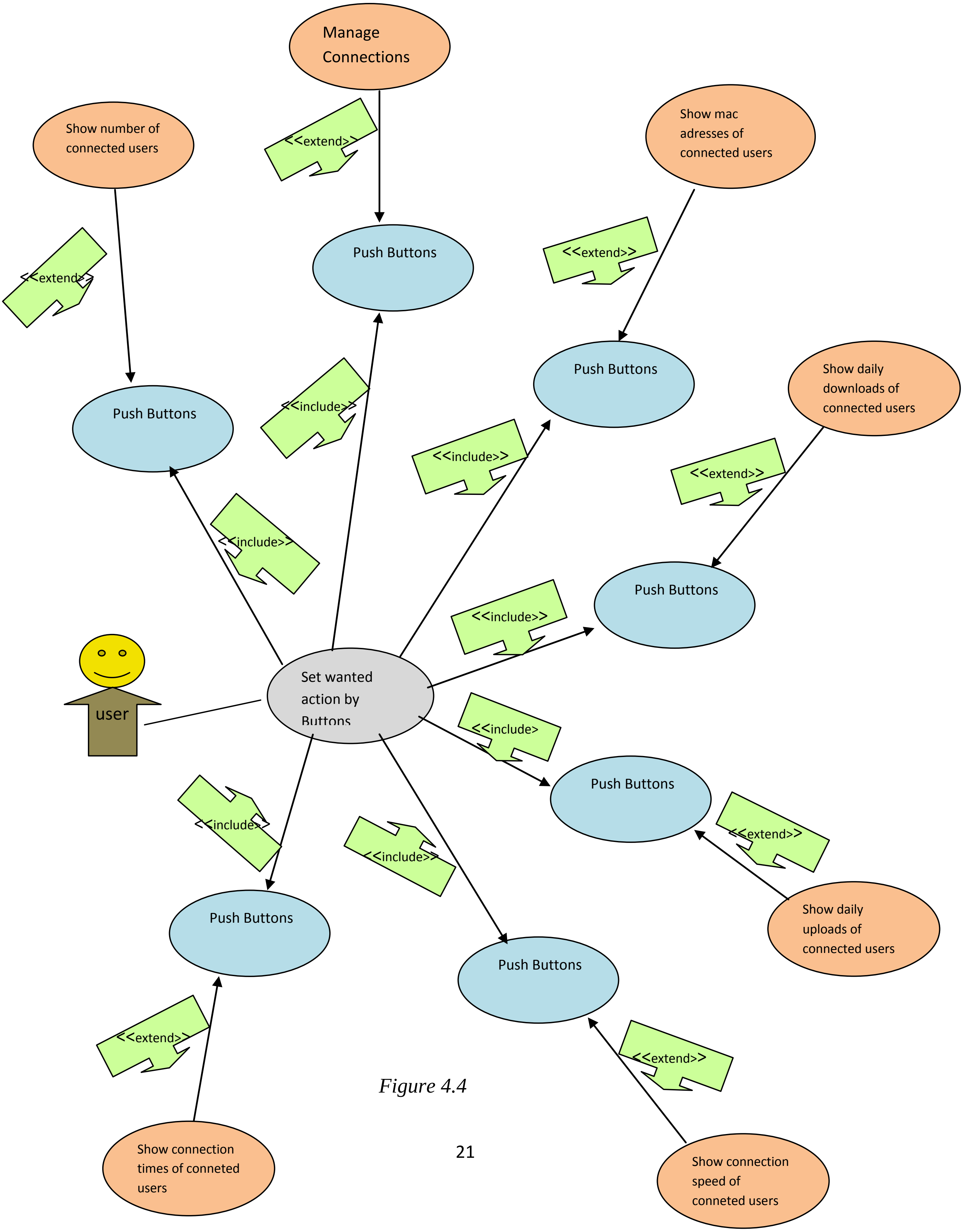


Figure 4.4

In our project user will select what he or she wants to do by pressing buttons of the board. Initially in the LCD display date will be seen. When user wants to do something he or she will press to the buttons and a selection menu will appear. Again pressing to the buttons user will select action to do. Then selected action will be done by AP400 with the help of pic 16f877. All this process is the goal of our project because user will see what he or she wants. So this process is very important part of out project.

- **Show Number of Connected Users :** By pressing buttons user can see the connected number of users when he or she wants. According to this information user can change the system settings. For example user can fix the data transmit rate to adequate value with network traffic. Also user can change the device frequency according to the number of connected users.
- **Show Mac Addresses of Connected Users :** User can see the mac addresses of connected users again pressing buttons of board. Mac addresses are unique so this information is very important. Because by mac addresses every user will be found very easily. User can do whatever he or she wants by only knowing the mac address of connected user. For example, only knowing mac addresses of connected user will be enough to block him or her.
- **Show Daily Downloads Of Connected Users :** User can see the daily downloads of the connected users by pressing buttons of board. Knowing daily downloads will be important due to use of project. For example in the internet cafes user may have quota on download size. So he or she can take precautions according to this information.
- **Show Daily Uploads Of Connected Users :** User can see the daily uploads of the connected users again by pressing buttons of the board. Like knowing daily downloads, knowing daily uploads will be important due to use of project. This information will be very important to the internet users who have quota on their accounts.
- **Show Connection Speed Of Connected Users :** User can see the connection speeds of connected user by pressing buttons of the board. This information will be beneficial to give feedback to connected user about his connection. If the connected users’ speed is low, then user can offer him to reset or repair his or her

connection. This process will help connected users to increase their benefits from device.

- **Show Total Connection Time Of Connected Users :** User can see the total connection time of the connected users by pushing buttons of the board. This info again can be useful due to use of project. This information will be very beneficial in big companies or internet cafes.

4.2) SEQUENCE DIAGRAMS

4.2.1) CHANGE SETTINGS MODULE

Change settings module consists of some sub-modules.

a) Change Settings: Change User Code Module

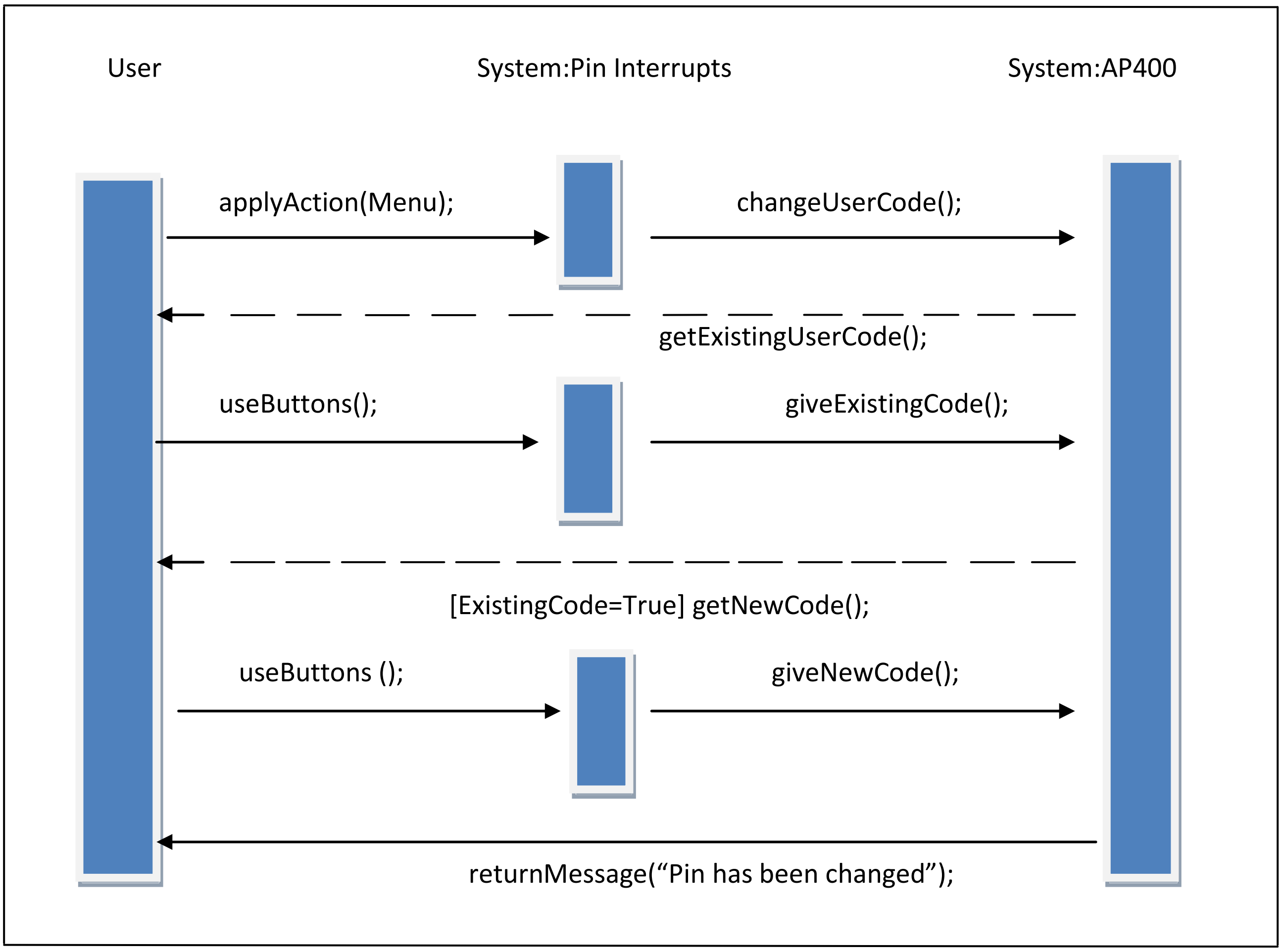


Figure 4.5

The above sequence diagram shows the interaction of the objects of our project when the user wants to change his pin code. Firstly, the user selects to change the pin code from the menu. When this action is selected, changeUserCode() function is called. Then, AP400 calls for the existing pin code, in order to check it, because only the owner can change the pin code. After that, by using buttons, the user gives an input of his existing pin code. Now, AP400 is ready to compare the input with the existing code. If this comparison returns true, a new input wanted. This input will be the new pin code. Again by using buttons, the user gives his new pin code. Lastly, AP400 sets the new pin code and returns a message.

b) Change Settings: Block Mac Adress Module

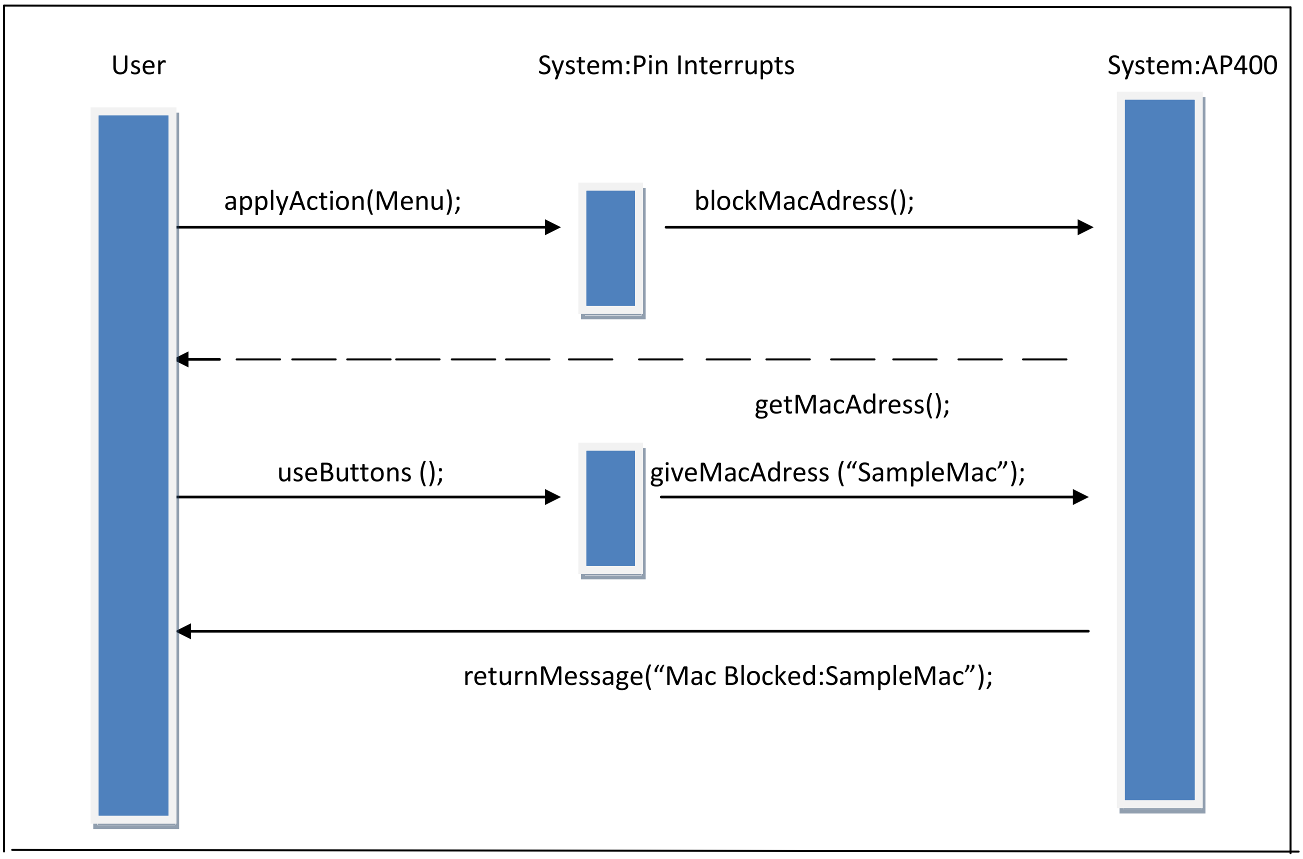


Figure 4.6

When the main user wants to block another user, he should block his mac address. In order to do that, the user selects the action from the menu. When AP400 warned with the function, blockMacAdress(), the user should type the mac address with the help of the buttons. When the user finishes typing the mac address, AP400 blocks this mac address and returns a message.

c) Change Settings: Change SSID Module

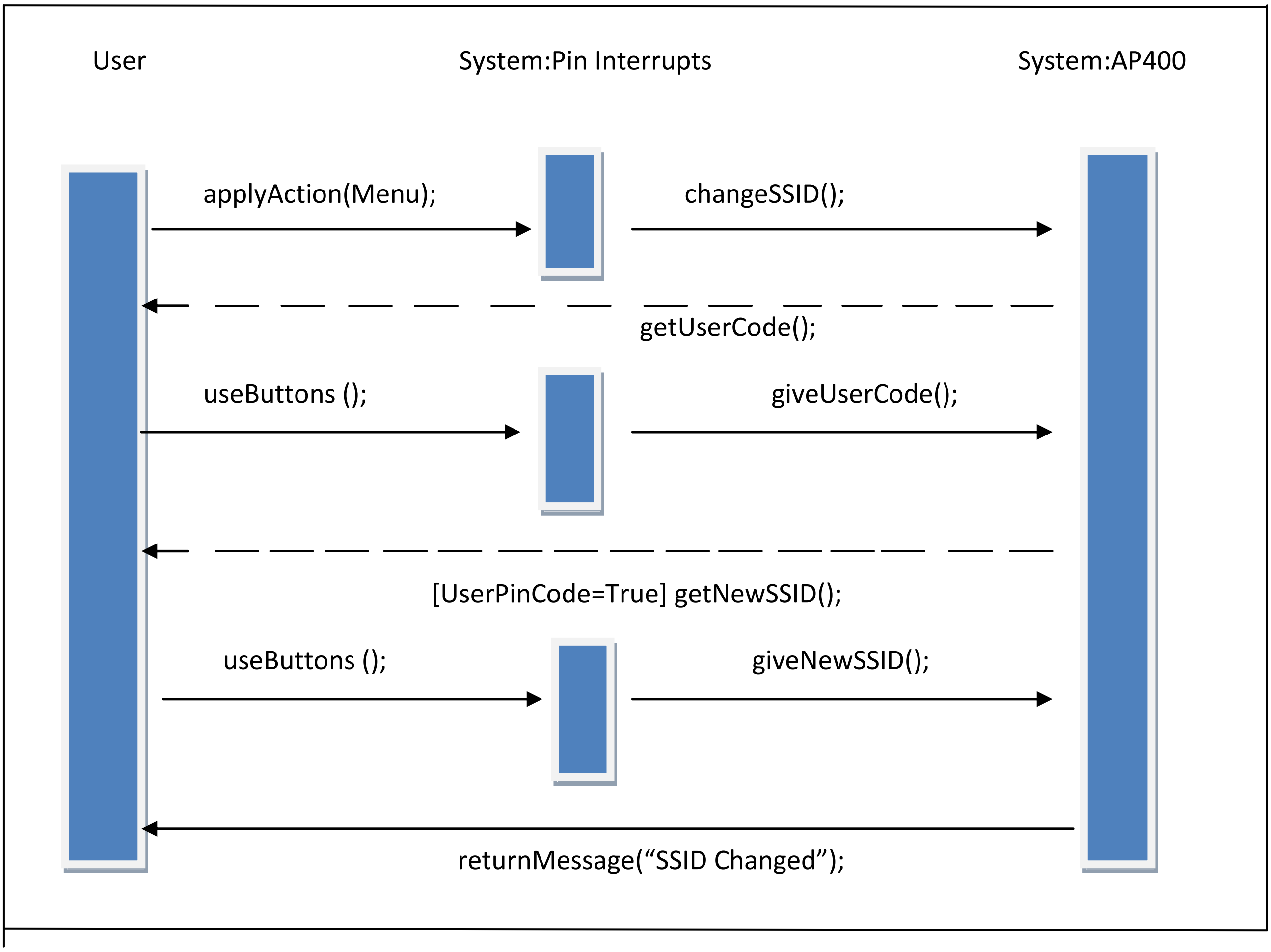


Figure 4.7

The above sequence diagram shows the sequential order of actions when the user wants to change the SSID. Firstly, the user selects to change the SSID code from the menu. When this action is selected, changeSSID() function is called. Then, AP400 calls for the user pin code, because only the owner can change the SSID. AP400 checks the user pin code. If it is true, AP400 calls for the new SSID. By using buttons, the user gives an input of wanted SSID. Then AP400 changes the SSID and returns a message.

d) Change Settings: Change Device Frequency Module

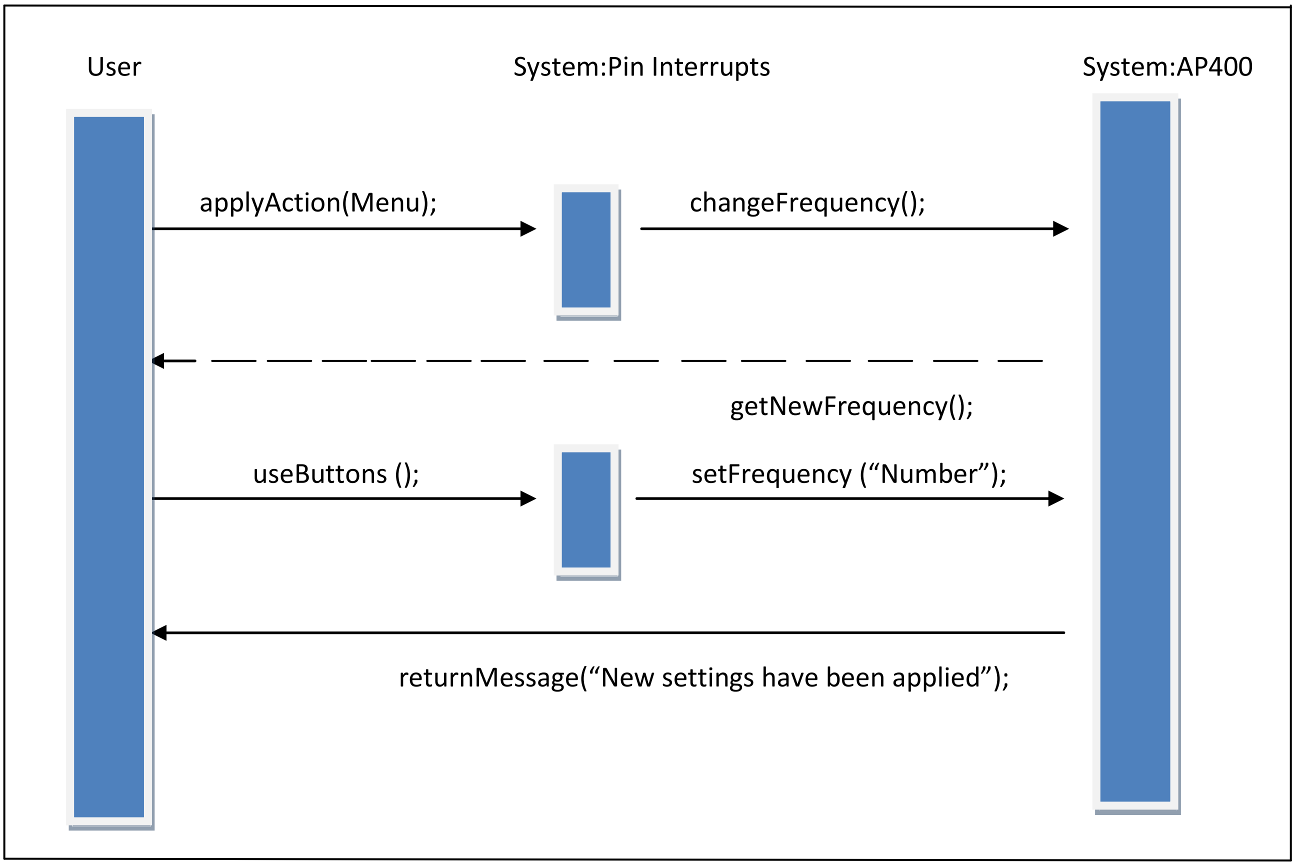


Figure 4.8

When the user wants to change frequency of the device, he will firstly choose this action from the menu. When the action is selected, changeFrequency() function is called. After that; AP400 asks for the new frequency value. By using buttons, the user will give an input of new frequency value. Then AP400 sets the new frequency value, and return a message.

e) Change Settings: Fix Data Transmit Rate Module

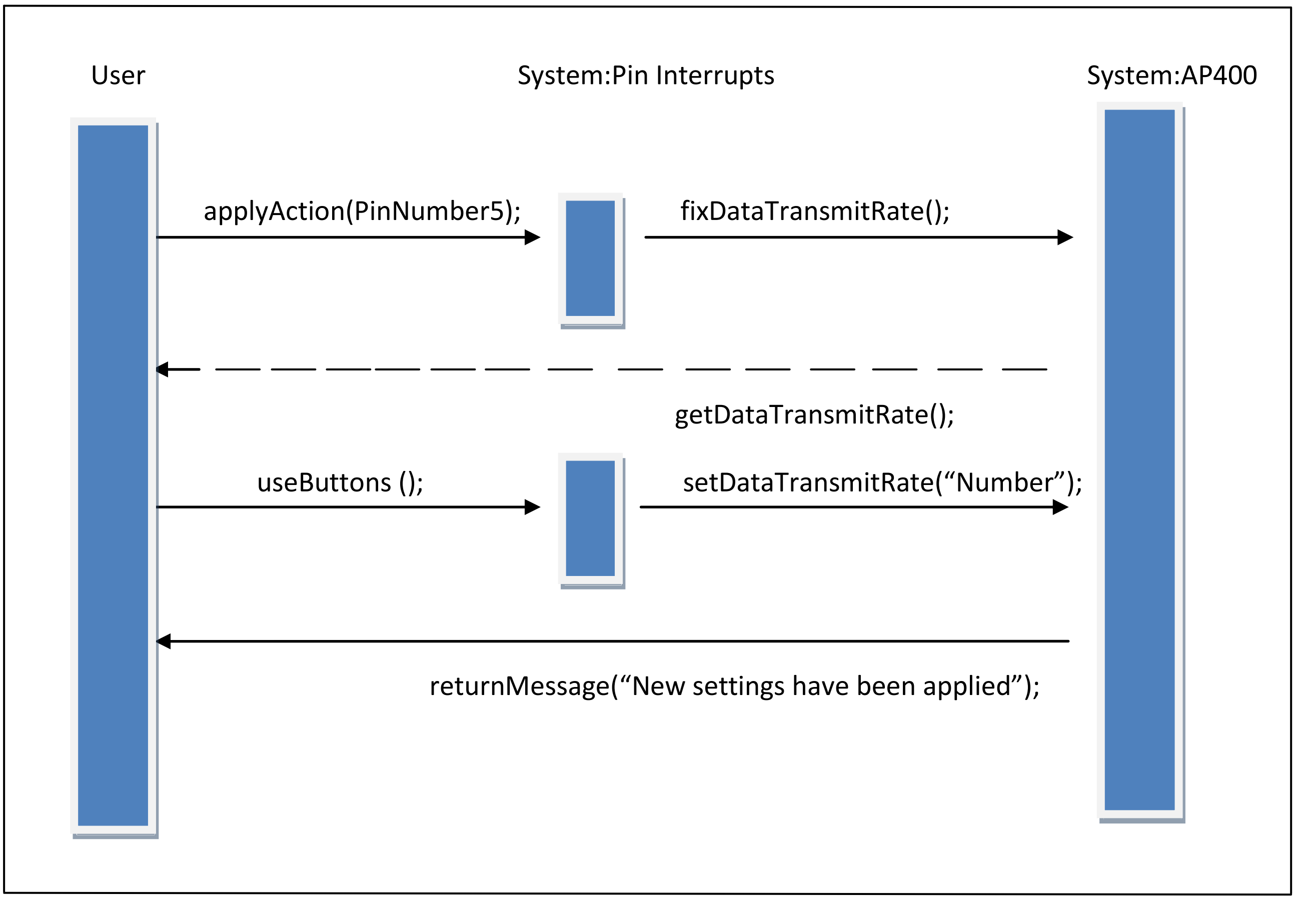


Figure 4.9

In order to fix the data transmit rate; the user should select the related action from the menu. Then, fixDataTransmitRate() function called. Then, AP400 returns with getDataTransmitRate() function. Now, it is expected from the user to enter the new data transmit rate value. If the user enters the new frequency value, AP400 returns a success message.

4.2.2) DISPLAY MODULE

Display module consists of some sub-modules.

a) **Display: Number of Connected Users Module**

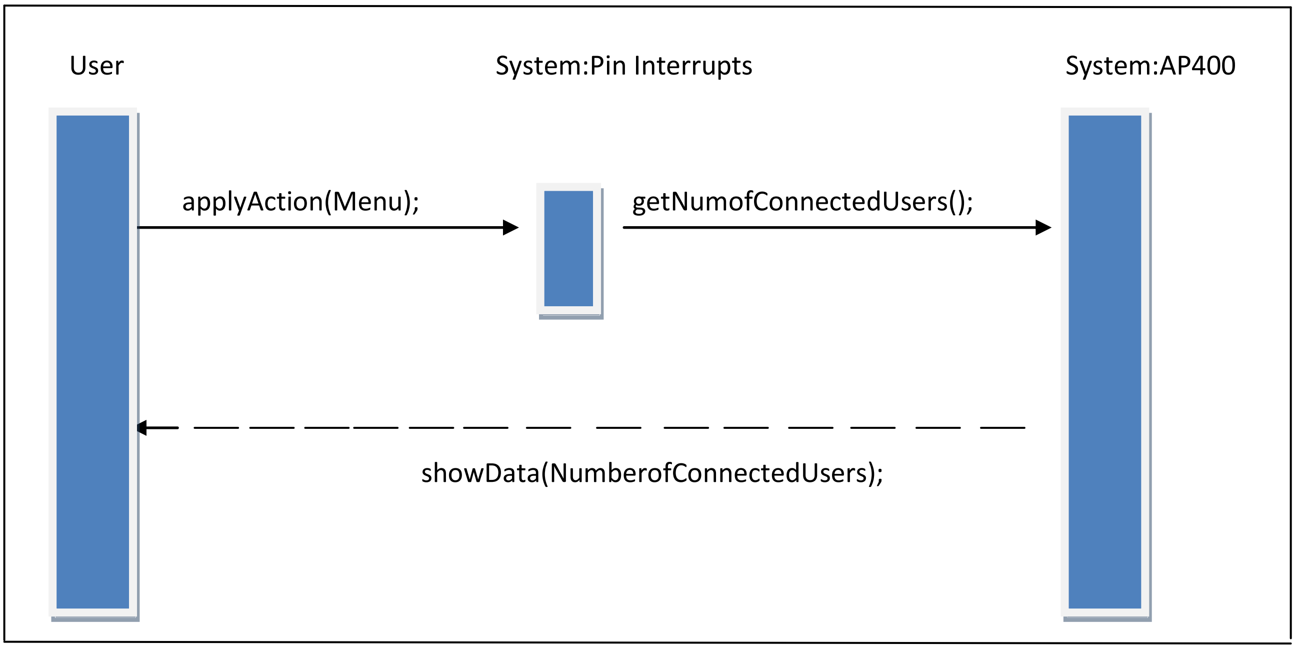


Figure 4.10

In order to see the number of connected users, the main user will set this action from the menu. Then, getNumofConnectedUsers() function is called. Then AP400 returns with the function showData() with the number of connected users.

b) **Display: Connection Manager Module**

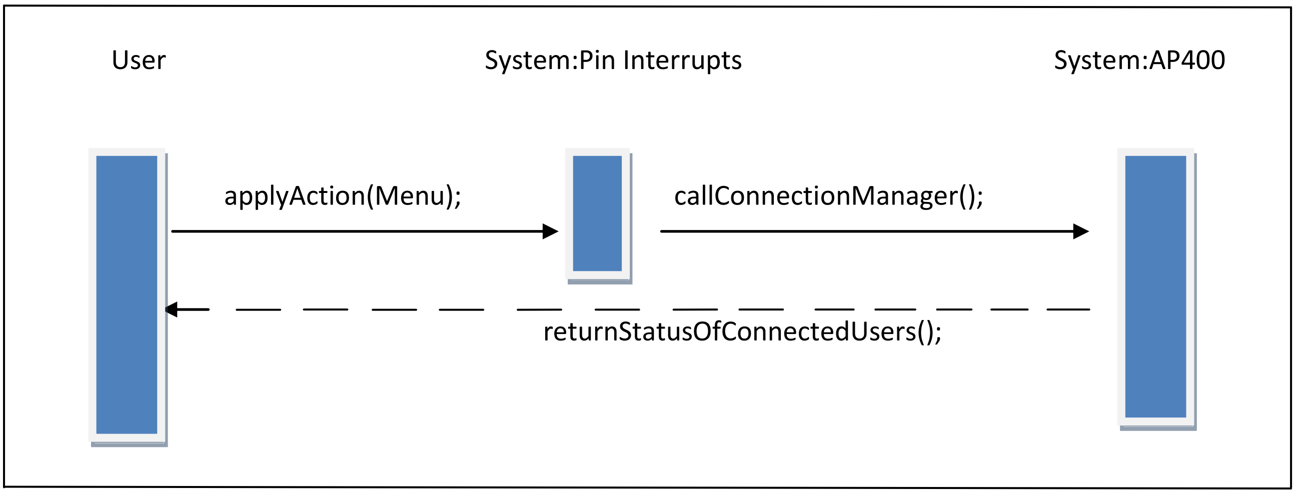


Figure 4.11

According to the diagram, when the user selects connection manager from the display menu, callConnectionManager() function is called. Then, AP400 returns a list of connected users.

c) Display: Mac Adresses of Connected Users Module

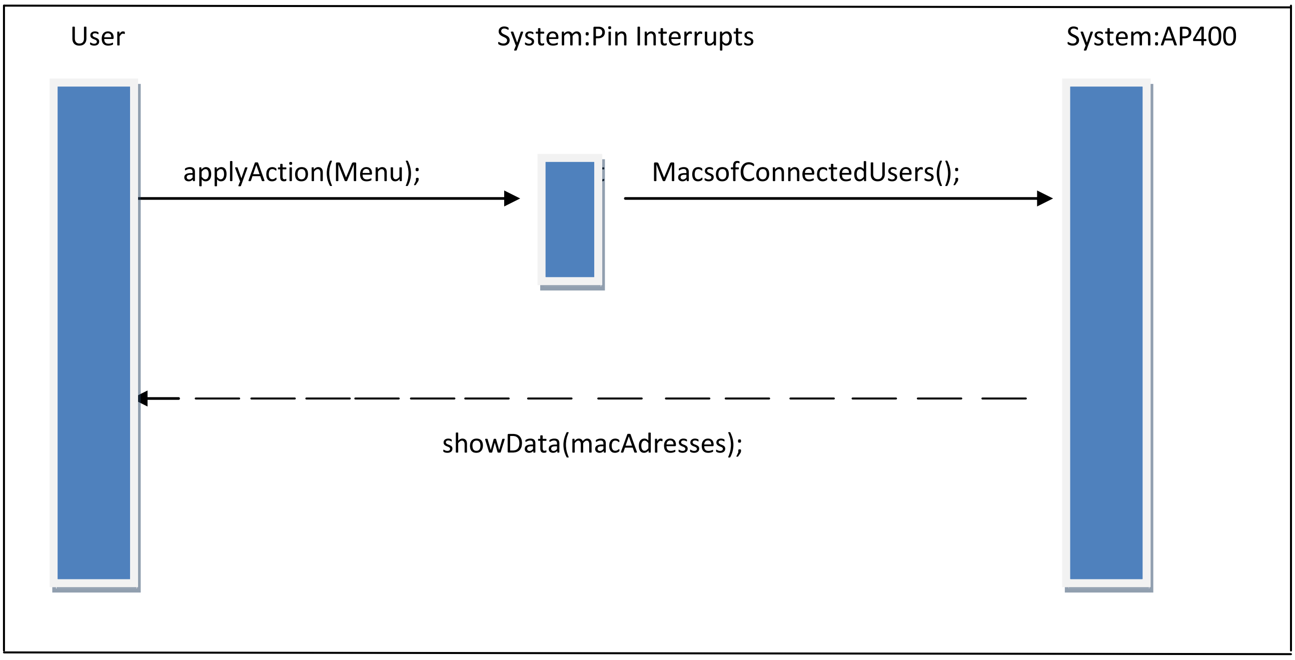


Figure 4.12

getMacsofConnectedUsers() function helps to get the mac addresses of the connected users to AP400. Here, showData() function returns the mac addresses of the connected users.

d) Display: Daily Downloads of Connected Users Module

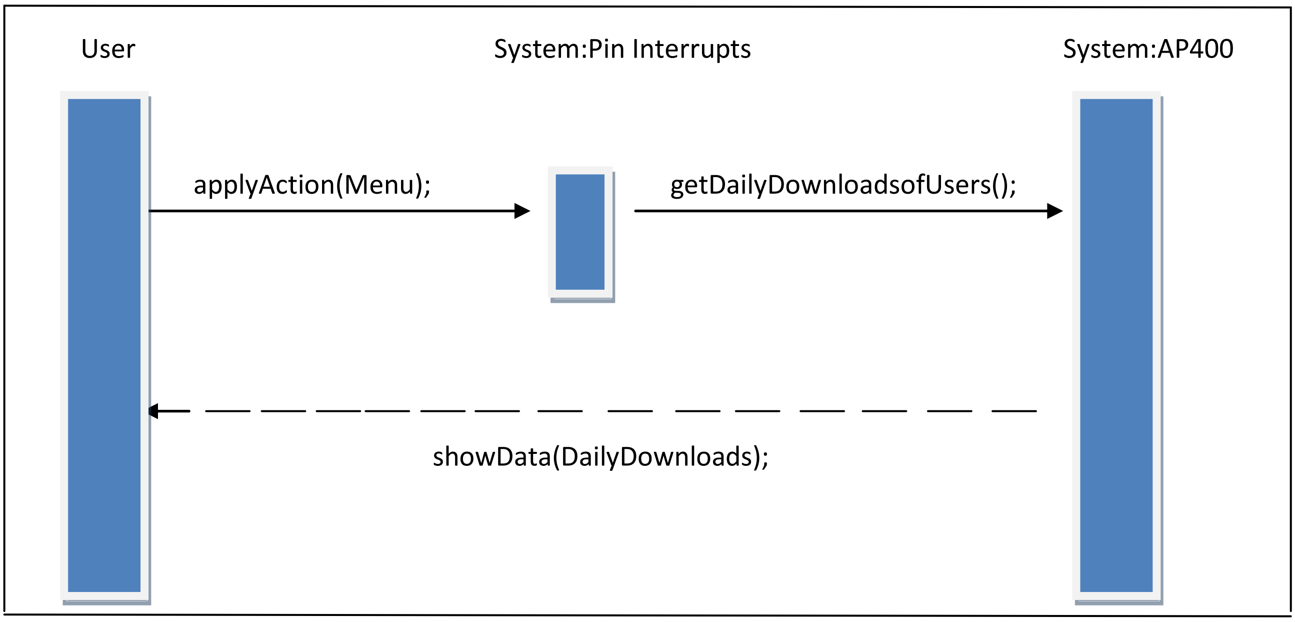


Figure 4.13

From the display menu; the user will select the action to see daily downloads of connected users to AP400. When AP400 is interrupted with the function getDailyDownloadsofUsers(), it will return the list of daily downloads.

e) Display: Daily Uploads of Connected Users Module

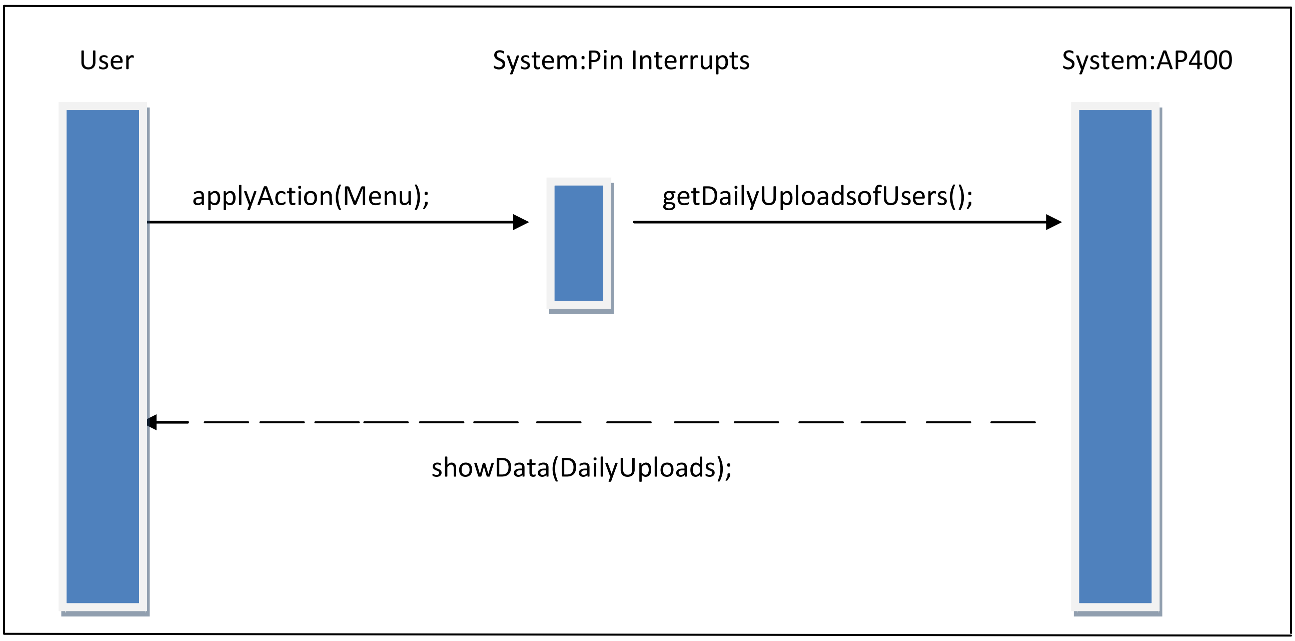


Figure 4.14

This module will work similar to the previous module. Only wanted data is daily uploads of the users.

f) Display: Connection Speeds of Connected Users Module

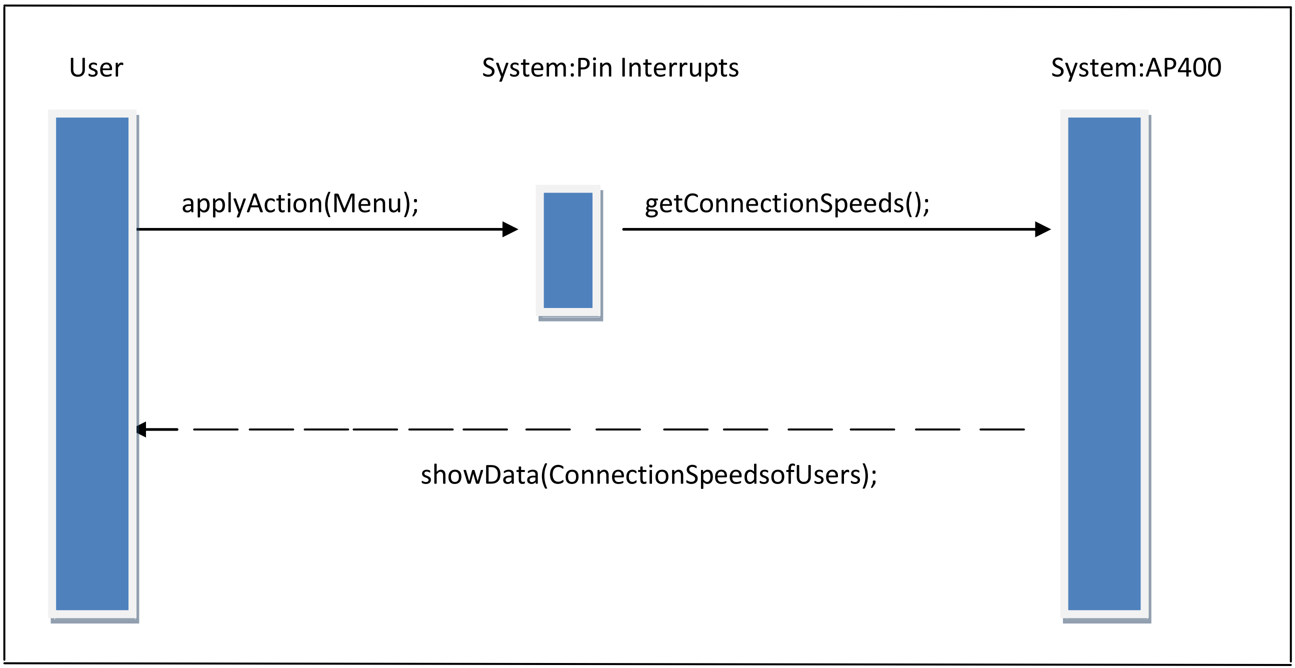


Figure 4.15

When the main user activates buttons to see the connection speeds of the users, getConnectionSpeeds() function is called. Then, AP400 returns connection speed of users by the help of the showData() function.

g) Display: Connection Times of Connected Users Module

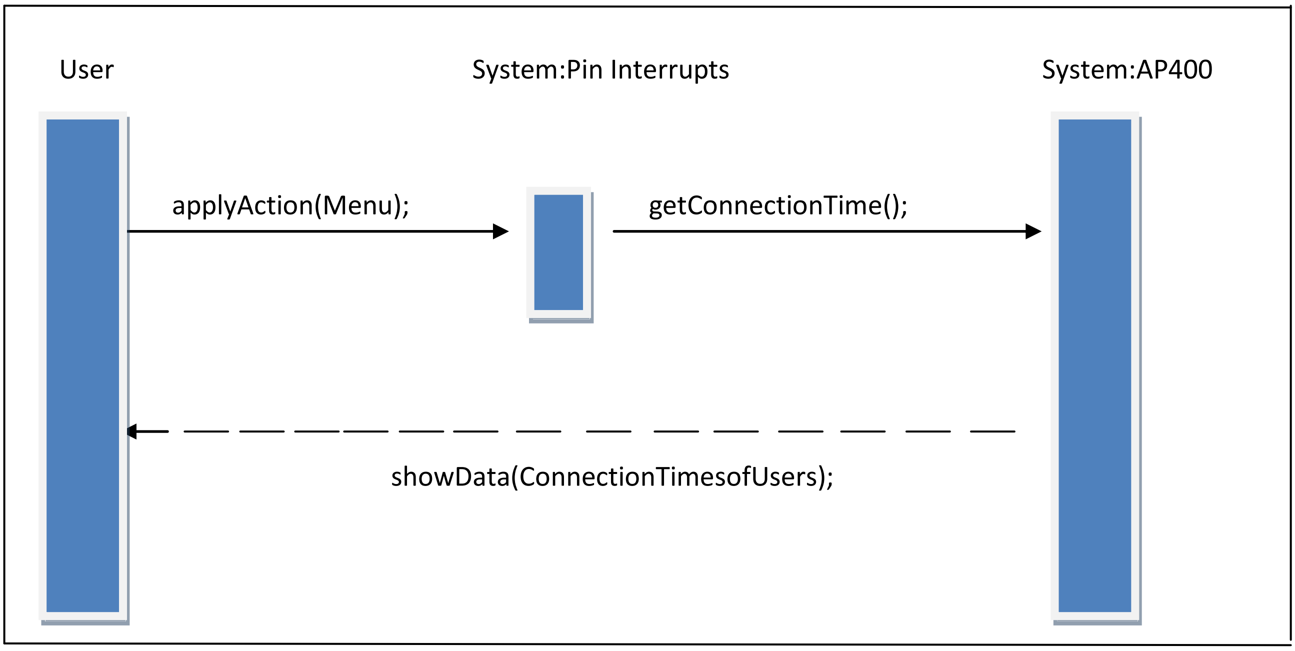


Figure 4.16

getConnectionTime() function will work for getting connection times of the connected users. When AP400 is interrupted by this function, it will return connection times of users by showData() function.

4.3) DATA FLOW DIAGRAMS

4.3.1) LEVEL 0

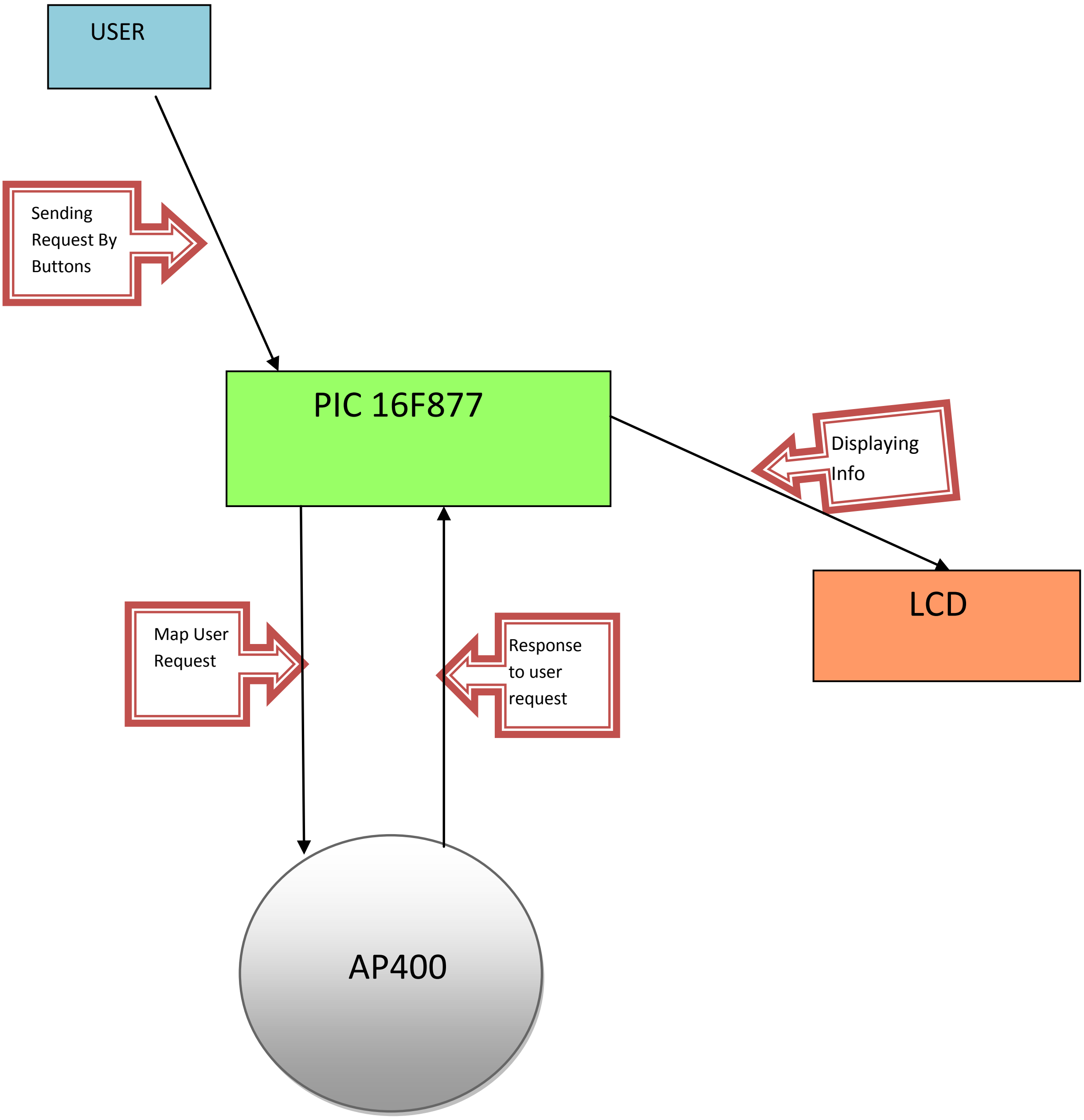


Figure 4.1

Data Dictionary:

Name: Sending requests by buttons

Output: User

Input: Pic 16f877

Description: User send request to Pic 16f877 by pressing buttons of the board. Its format is electrical signal.

Name: Map user request

Output: Pic 16f877

Input: AP400

Description: Pic takes user requests and then maps them to AP400. It is in RS232 serial port connection format.

Name: Response to user requests

Output: AP400

Input: Pic 16f877

Description: AP400 sends the results of user request to the pic 16f877. It is in RS232 serial port connection.

Name: Displaying Info

Output: Pic 16f877

Input: LCD

Description: Pic 16f877 sends the data which it receives from AP400 to the LCD display. Its format is electrical signals.

4.3.2) LEVEL 1

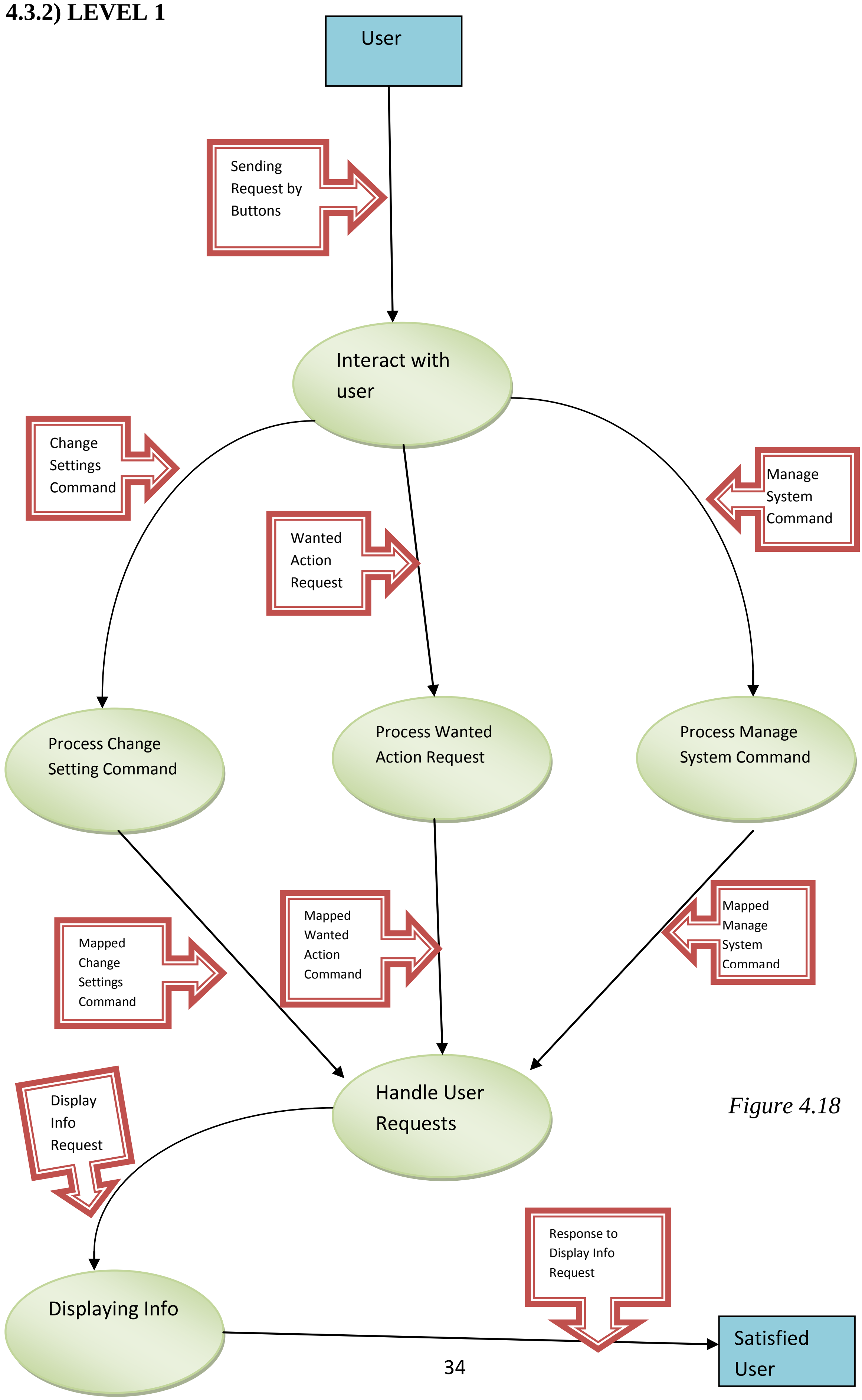

User
Sending Request by Buttons
Interact with user
Change Settings Command
Wanted Action Request
Manage System Command
Process Change Setting Command
Process Wanted Action Request
Process Manage System Command
Mapped Change Settings Command
Mapped Wanted Action Command
Mapped Manage System Command
Handle User Requests
Display Info Request
Displaying Info
Response to Display Info Request
Satisfied User

Figure 4.18

Data Dictionary:

Name: Change settings command

Output: Interaction with user

Input: Process change settings command

Description: The output of the interaction will go to the “process change settings command” and this process will evaluate this info. Its format is electrical signal.

Name: Wanted action request

Output: Interaction with user

Input: Process wanted action request

Description: The output of the interaction will go to the “process wanted action request” and this process will evaluate this info. Its format is electrical signal.

Name: Manage system command

Output: Interaction with user

Input: Process manage system command

Description: The output of the interaction will go to the “process manage system command” and this process will evaluate this info. Its format is electrical signal.

Name: Mapped change settings command

Output: Process change setting command

Input: Handle user request

Description: The mapped request of the user will go to the “handle user request” and this process will adjust the settings as user wants. Its format is electrical signal.

Name: Mapped wanted action command

Output: Process wanted action request

Input: Handle user request

Description: The mapped request of the user will go to the “handle user request” and this process will send the info to the LCD display. Its format is electrical signal.

Name: Mapped manage system command

Output: Process manage system command

Input: Handle user request

Description: The mapped manage system command will go to the “handle user request” and this process will decide the state of the system. Its format is electrical signal.

Name: Display info request

Output: Handle user request

Input: Display info

Description: The info handled by “Handle user request” will be send to the “Display info”. Then this process will display the info on the LCD display. Its format is electrical signal.

Name: Response to display info request

Output: Display info

Input: Satisfied user

Description: The info received by “Display info” will be displayed in LCD display. Its format is electrical signal.

4.3.3) LEVEL 2

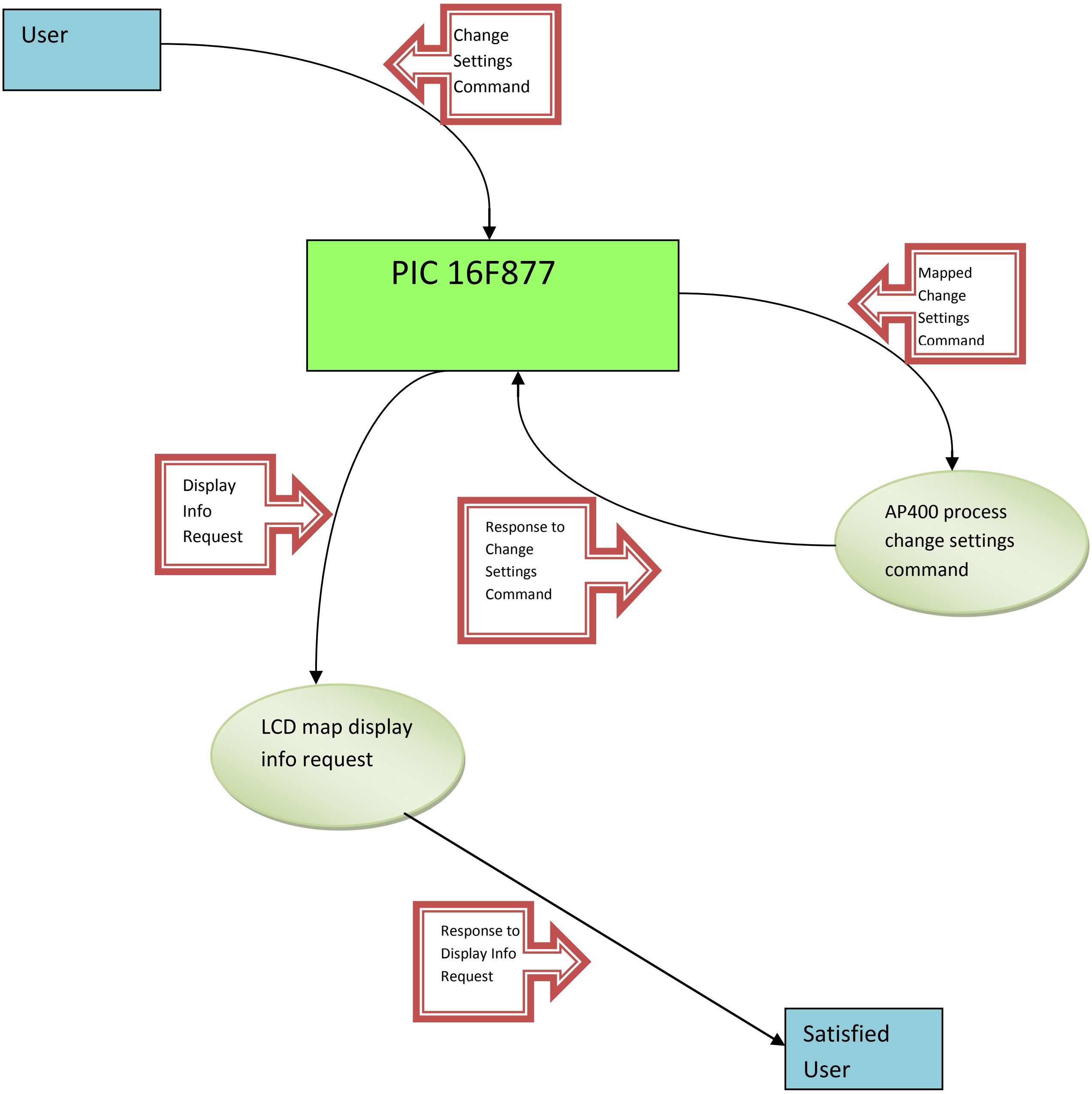

User
Change Settings Command
PIC 16F877
Mapped Change Settings Command
Display Info Request
Response to Change Settings Command
AP400 process change settings command
LCD map display info request
Response to Display Info Request
Satisfied User

Figure 4.19

Data Dictionary:

Name: Change settings command

Output: User

Input: Pic 16f877

Description: User send request to Pic 16f877 by pressing buttons of the board. Its format is electrical signal.

Name: Map change setting command

Output: Pic 16f877

Input: AP400 process change setting command

Description: Pic 16f877 will take the request of the user and then it will map to the “AP400 process change setting command”. Its format will be electrical signal.

Name: Response to change setting command

Output: AP400 process change setting command

Input: Pic 16f877

Description: AP400 will process on the request it receive from pic 16f877 and than send response to this request to the pic 16f877. Its format is electrical signal.

Name: Display info request

Output: Pic 16f877

Input: LCD map display info request

Description: The request received from “pic 16f877” will be processed by LCD display and it will be ready to seen on screen.

Name: Response to display info request

Output: LCD map display info request

Input: Satisfied user

Description: The info received by “Display info” will be displayed in LCD display .Its format is electrical signal.

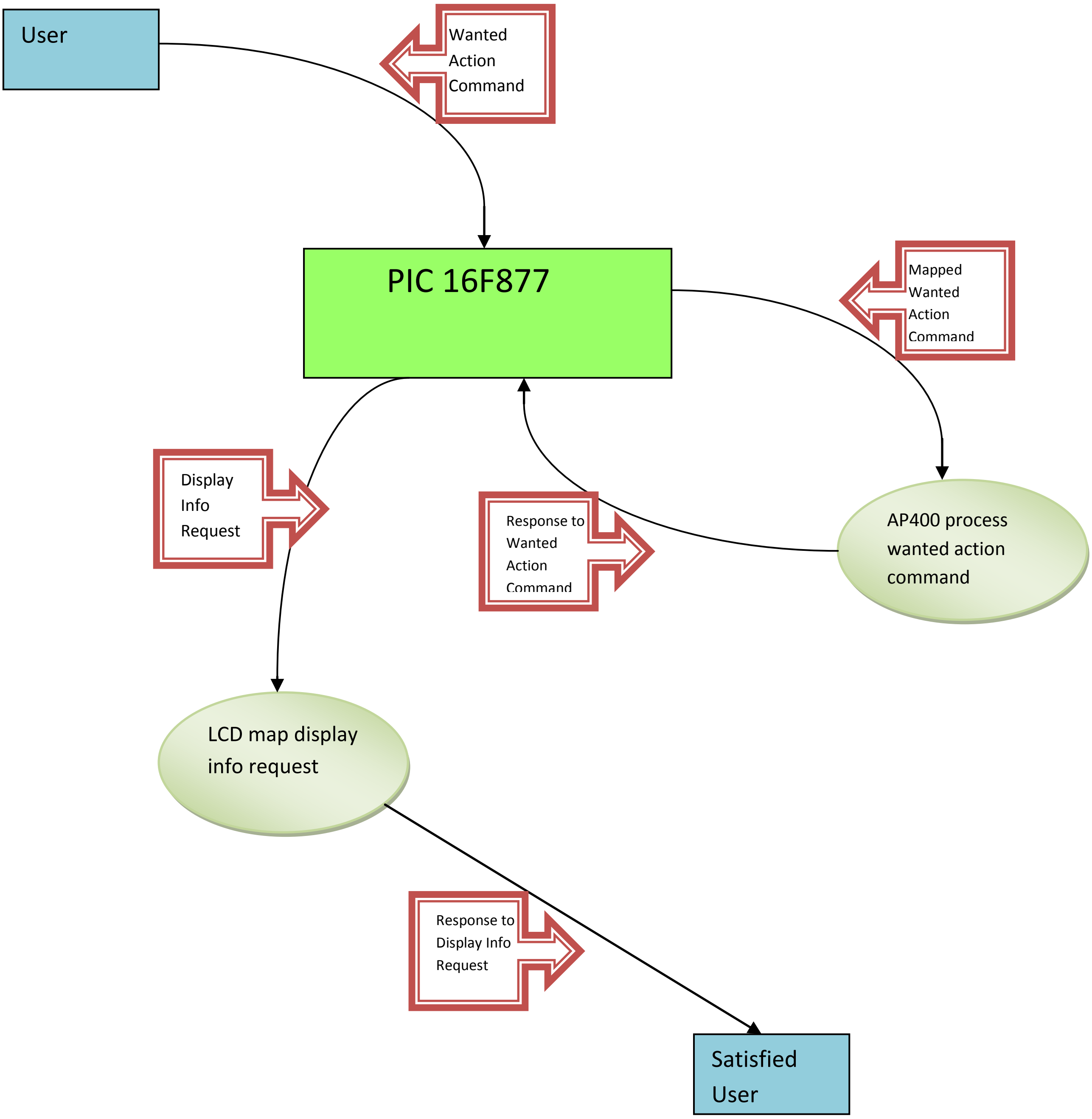

User
Wanted Action Command
PIC 16F877
Mapped Wanted Action Command
Display Info Request
Response to Wanted Action Command
AP400 process wanted action command
LCD map display info request
Response to Display Info Request
Satisfied User

Figure 4.20

Data Dictionary:

Name: Wanted action request

Output: User

Input: Pic 16f877

Description: User send request of what he or she wants to the Pic 16f877 by pressing buttons of the board . Its format is electrical signal.

Name: Map wanted action command

Output: Pic 16f877

Input: AP400 process wanted action command

Description: Pic 16f877 will take the request of the user and then it will map to the “AP400 process wanted action command”. Its format will be electrical signal.

Name: Response to wanted action command

Output: AP400 process wanted action command

Input: Pic 16f877

Description: AP400 will process on the request it receive from pic 16f877 and than send response to this request to the pic 16f877. Its format is electrical signal.

Name: Display info request

Output: Pic 16f877

Input: LCD map display info request

Description: The request received from “pic 16f877” will be processed by LCD display and it will be ready to seen on screen.

Name: Response to display info request

Output: LCD map display info request

Input: Satisfied user

Description: The info received by “Display info” will be displayed in LCD display .Its format is electrical signal.

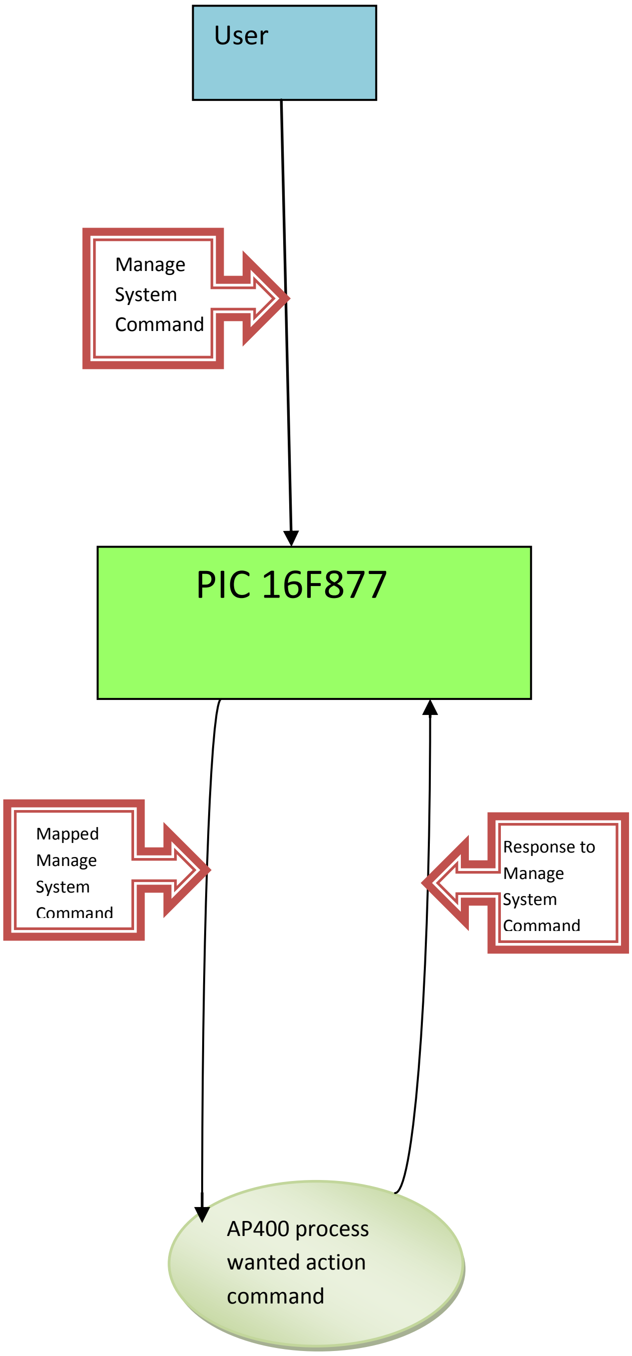

User
Manage System Command
PIC 16F877
Mapped Manage System Command
Response to Manage System Command
AP400 process wanted action command

Figure 4. 21

Data Dictionary:

Name: Manage system command

Output: User

Input: Pic 16f877

Description: User send request of what he or she wants to the Pic 16f877 by pressing buttons of the board. Its format is electrical signal.

Name: Map change system command

Output: Pic 16f877

Input: AP400 process manage system command

Description: Pic 16f877 will take the request of the user and then it will map to the "AP400 process wanted action command". Its format will be electrical signal.

Name: Response to manage system command

Output: AP400 process manage system command

Input: Pic 16f877

Description: AP400 will process on the request it receive from pic 16f877 and than send response to this request to the pic 16f877. Its format is electrical signal.

5) USER INTERFACE

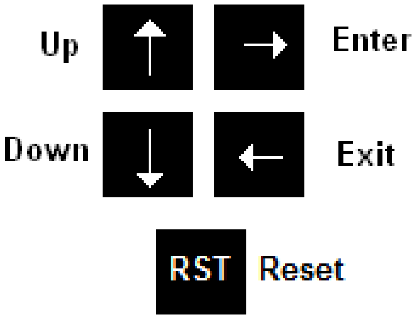


Figure 5.1

There will be 5 buttons on the board. Two of them will be used for going up and down in menus. The two other buttons will be used for entering to a menu and exiting from a menu. And the remaining button will be used for resetting the LCD display.

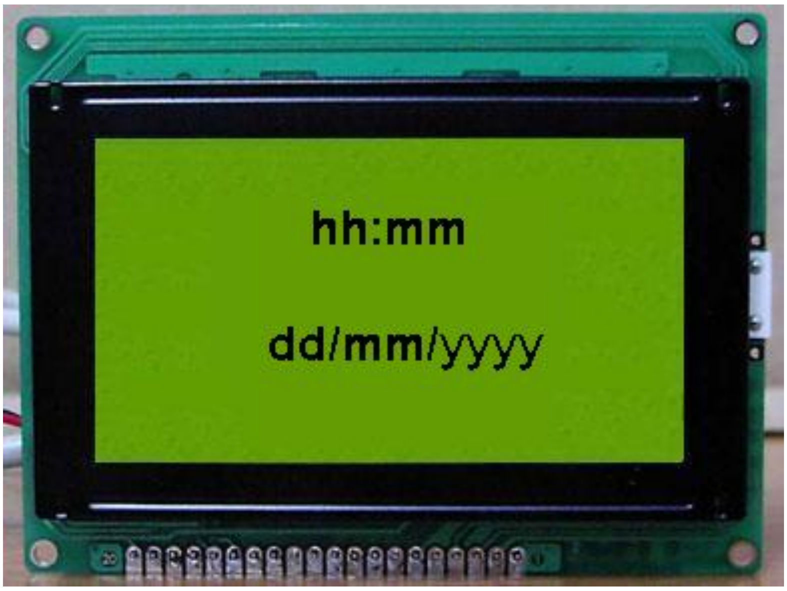


Figure 5.2

When the user has not pressed any button yet, the time and the date is going to be displayed as in figure 5.2 until the user presses any button on the board.

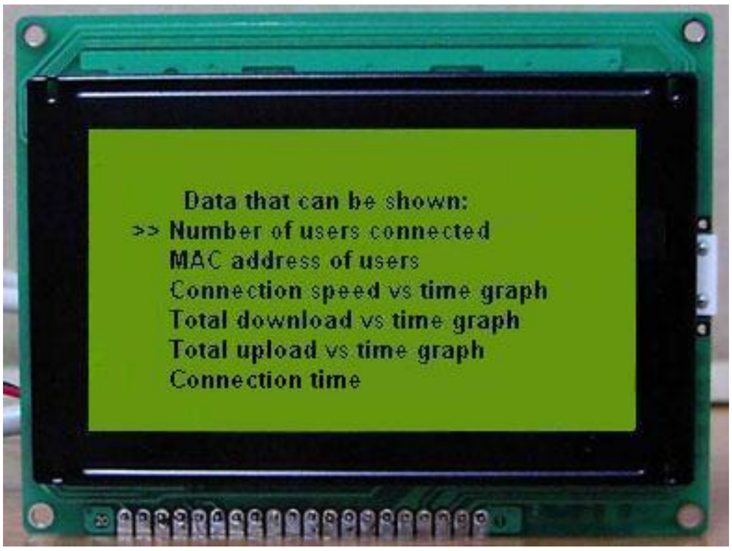
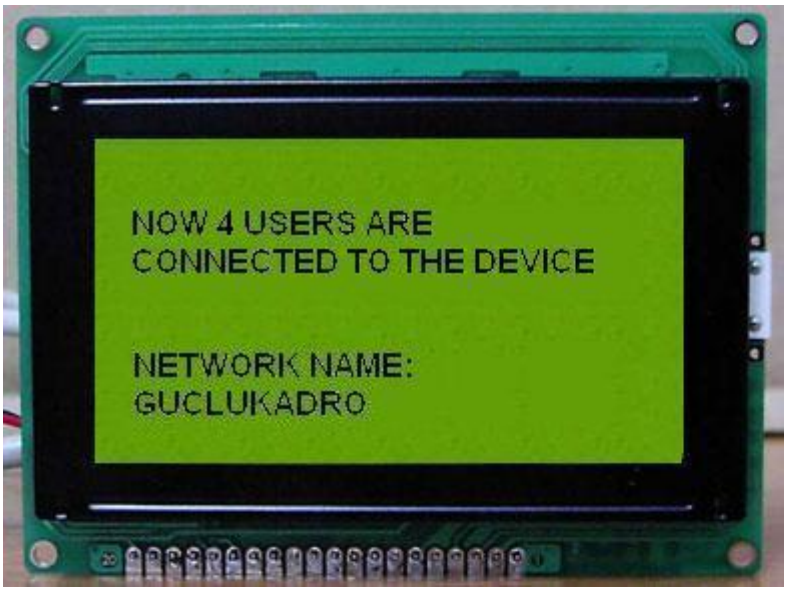

Data that can be shown:
>> Number of users connected
MAC address of users
Connection speed vs time graph
Total download vs time graph
Total upload vs time graph
Connection time

Figure 5.3

When the user presses any button on the board the screen is going to display the list of data that can be shown as in figure 5.3. There are some other actions such as change PIN code and SSID. These items will be shown in the second page when the user presses the down button. If the user selects to change the PIN code, after entering existing PIN code, he/she will enter the new PIN code. Typing will be similar to typing in mobile phones. The character will change with exit button. The cursor will move according to the time interval between pressing buttons. Using the enter button the cursor will move between the typed characters in order to be able to correct the mistyped ones. Also there will be cancel and finish options in this menu. The user will be able to switch between the options with up and down buttons. Enter button will be used for entering to these options. Changing the SSID process will be similar to this process.

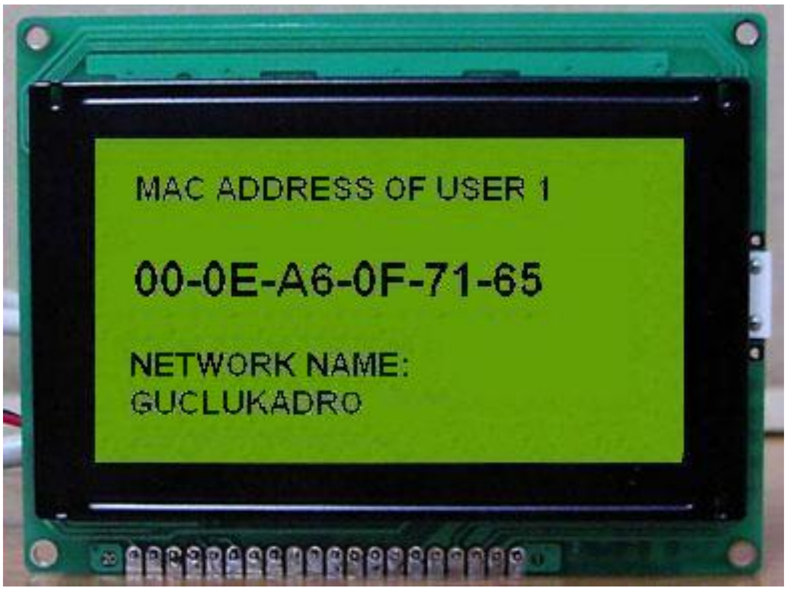


NOW 4 USERS ARE
CONNECTED TO THE DEVICE

NETWORK NAME:
GUCLUKADRO

Figure 5.4

When the user selects the number of users connected and presses the enter button, the number of clients who are currently using the device is going to appear on the LCD display with the name of the network name as in figure 5.4. When the user presses the exit button, it will return to the main menu.



MAC ADDRESS OF USER 1

00-0E-A6-0F-71-65

NETWORK NAME:
GUCLUKADRO

Figure 5.5

When the user selects the show MAC address choice from the list and presses the enter button MAC address of the clients that are currently connected to the device is going to be displayed with the network name as in figure 5.5. The MAC address of the other clients is going to be displayed on the LCD display if the user presses the up/down button. When the user presses the exit button, it will return to the main menu.

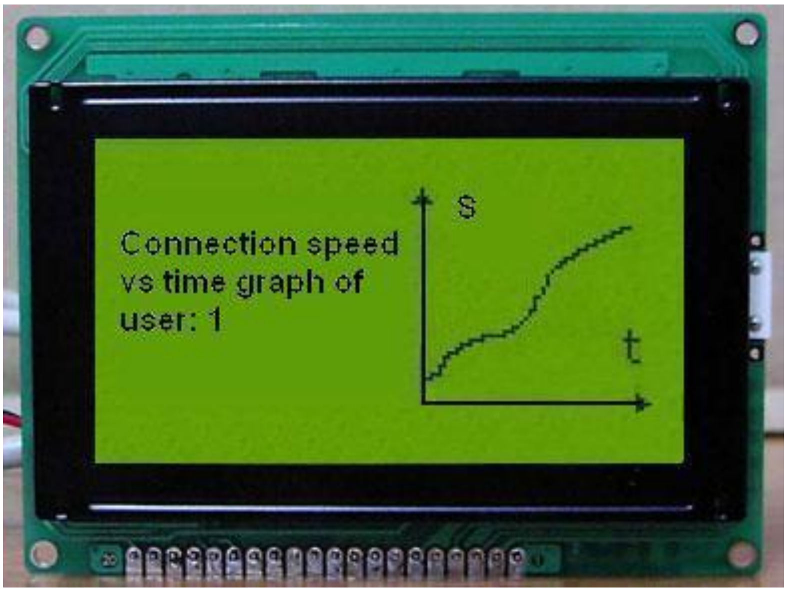


Figure 5.6

When the user selects the show connection speed choice from the list and presses the enter button connection speed vs. time graph of each client that is connected to the device is going to be displayed as in figure 5.6. The graph of the other clients is going to be displayed on the LCD display if the user presses the up/down button. When the user presses the exit button, it will return to the main menu.

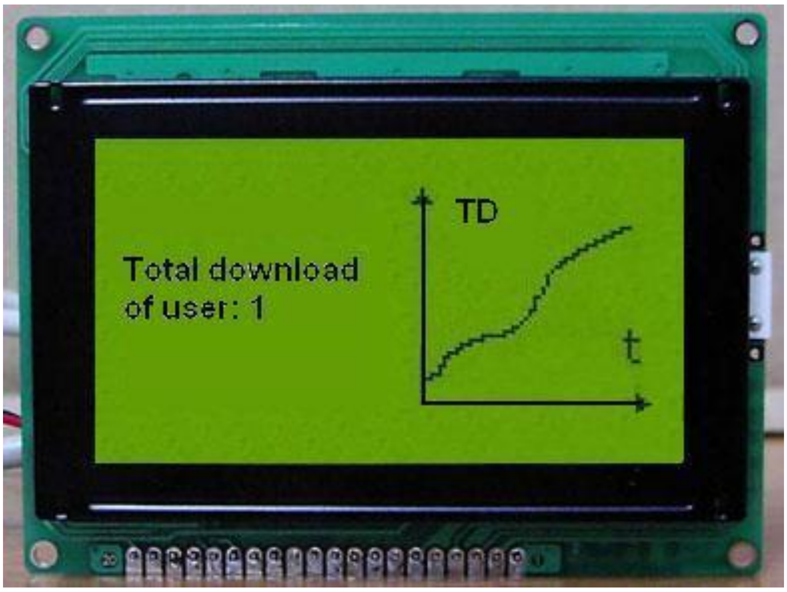


Figure 5.7

When the user selects the show total download of the users choice from the list and presses the enter button total download vs. time graph of each client that is connected to the device is going to be displayed as in figure 5.7. The user is able to see the graph of other clients by pressing the up/down button. When the user presses the exit button, it will return to the main menu.

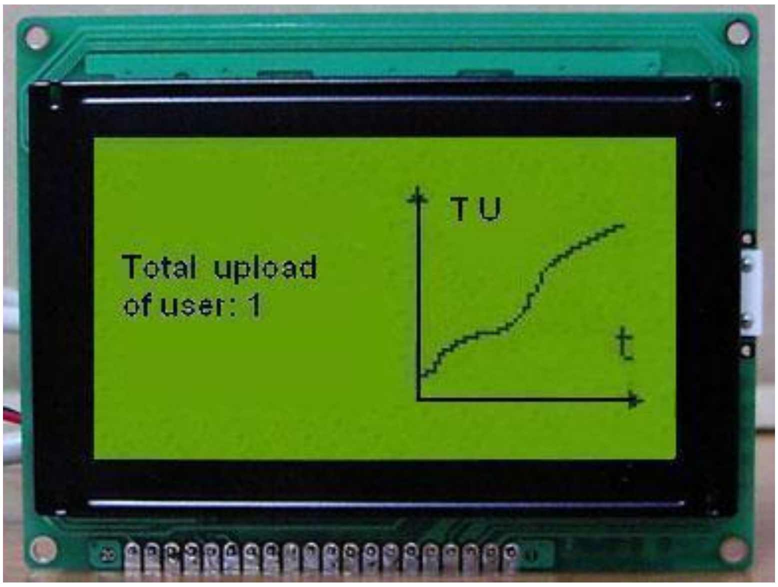
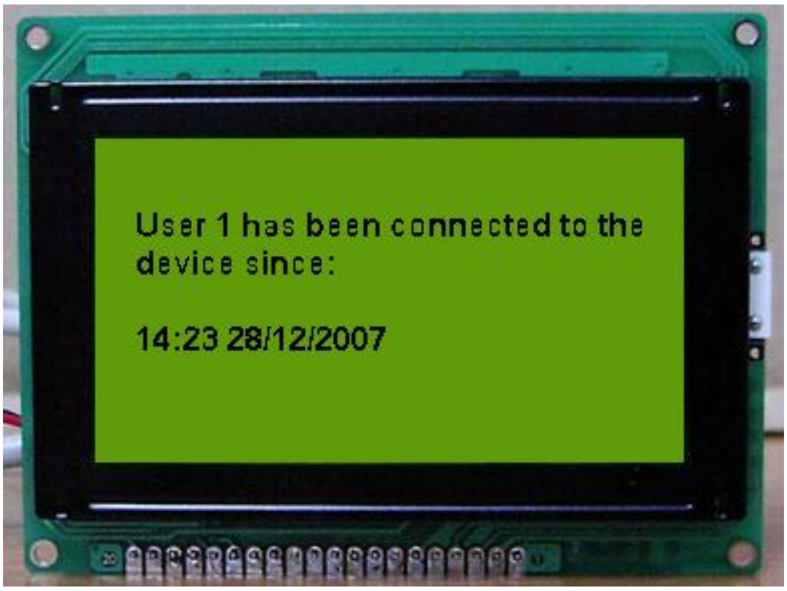


Figure 5.8

When the user selects the show total upload of the users choice from the list and presses the enter button total upload vs. time graph of each client that are connected to the device is going to be displayed as in figure 5.8. The user is able to see the graph of other clients by pressing the up/down button. The user can also return to the main menu by pressing the exit button.



User 1 has been connected to the device since:

14:23 28/12/2007

Figure 5.9

When the user selects the show connection time of the users choice from the list and presses the enter button the connection time of each client that are connected to the device is going to be displayed as in figure 5.9. The connection time of the other clients can be displayed on the LCD display by up/down button. The user can return to the main menu by pressing the exit button.

6) CONCLUSION

This is the final design report of our project that we showed our latest designs of our prototype. We clearly mentioned our software and hardware designs, and also we stated our hardware requirements. In addition, we pointed out our Use-Case, Sequence and Data Flow Diagrams in modeling part. Our designs are the improved ones of the designs that we showed in the Initial Design Report. We think that, these will be sufficient to complete the version 1.0 of our product. Also, they are the basics of future version of our product. In other words, if we want to extend our project in 2nd semester, we will use these designs as a base and add new features.

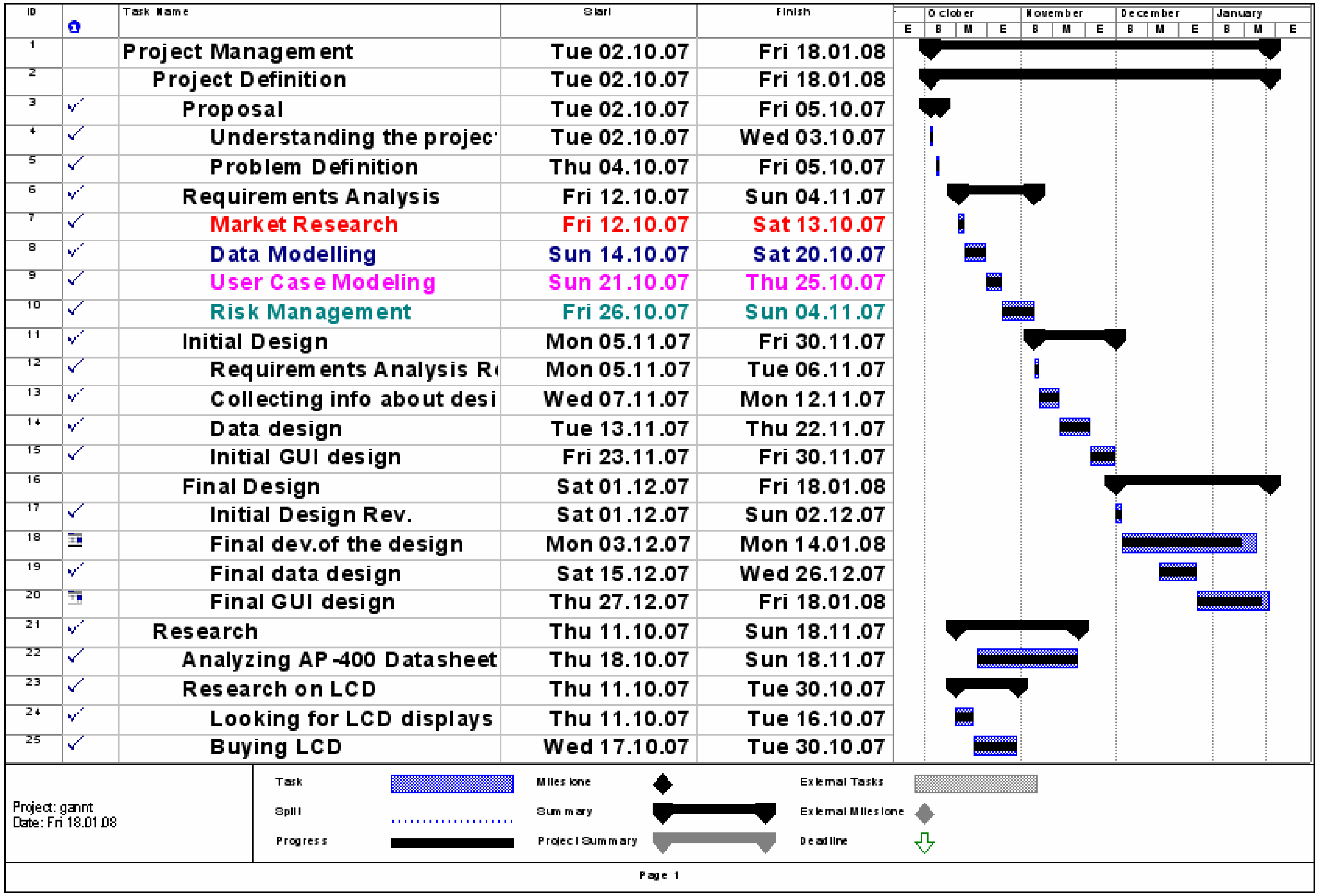
7) GANNT CHART

🟥 BERK ÇAKIR

🟦 RECEP ALABAŞ

🟪 ÖZER DURMAZ

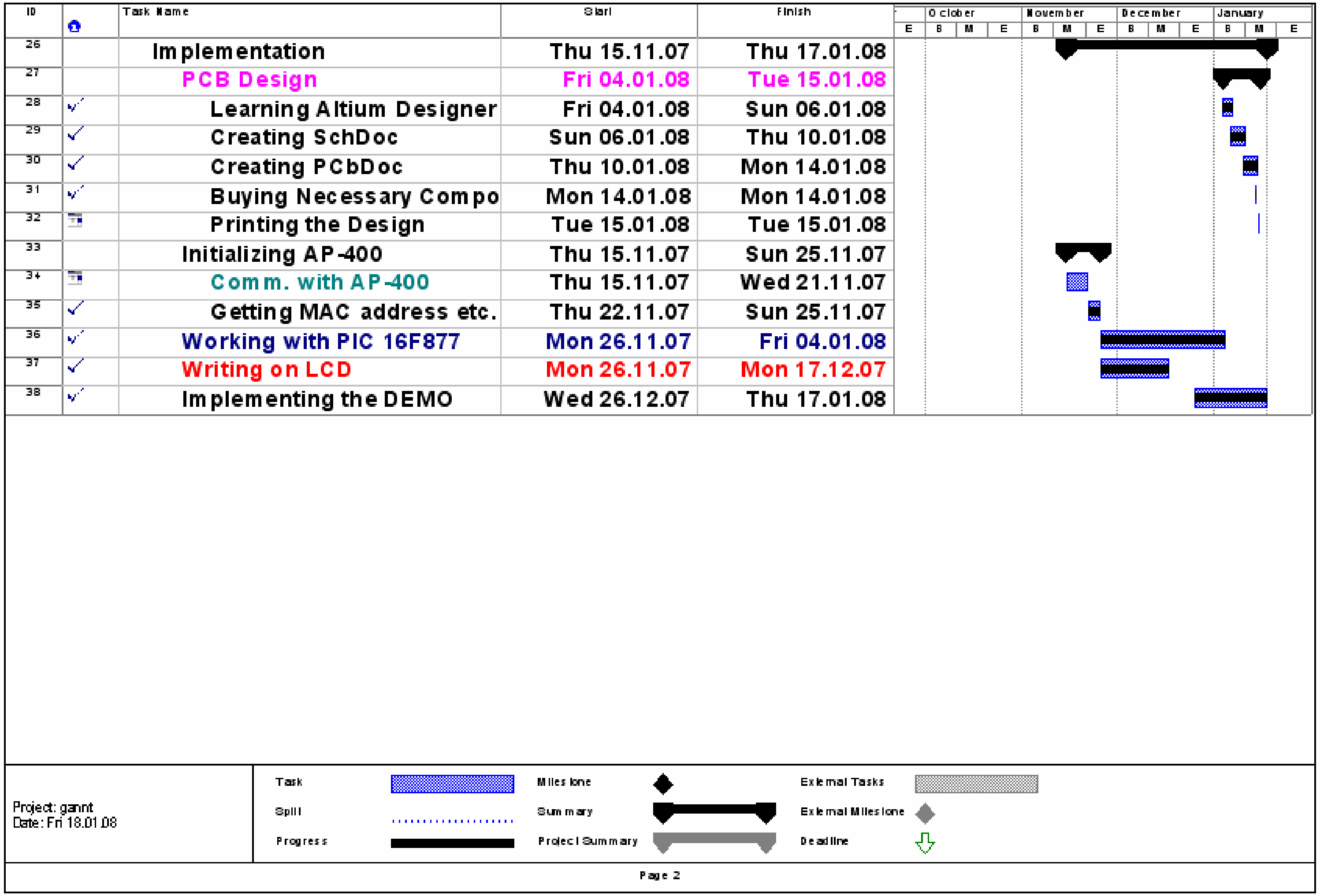
🟩 YAŞAT K. BİLGİN



ID		Task Name	Start	Finish
1		Project Management	Tue 02.10.07	Fri 18.01.08
2		Project Definition	Tue 02.10.07	Fri 18.01.08
3	✓	Proposal	Tue 02.10.07	Fri 05.10.07
4	✓	Understanding the projec	Tue 02.10.07	Wed 03.10.07
5	✓	Problem Definition	Thu 04.10.07	Fri 05.10.07
6	✓	Requirements Analysis	Fri 12.10.07	Sun 04.11.07
7	✓	Market Research	Fri 12.10.07	Sat 13.10.07
8	✓	Data Modelling	Sun 14.10.07	Sat 20.10.07
9	✓	User Case Modeling	Sun 21.10.07	Thu 25.10.07
10	✓	Risk Management	Fri 26.10.07	Sun 04.11.07
11	✓	Initial Design	Mon 05.11.07	Fri 30.11.07
12	✓	Requirements Analysis R	Mon 05.11.07	Tue 06.11.07
13	✓	Collecting info about desi	Wed 07.11.07	Mon 12.11.07
14	✓	Data design	Tue 13.11.07	Thu 22.11.07
15	✓	Initial GUI design	Fri 23.11.07	Fri 30.11.07
16		Final Design	Sat 01.12.07	Fri 18.01.08
17	✓	Initial Design Rev.	Sat 01.12.07	Sun 02.12.07
18	▤	Final dev.of the design	Mon 03.12.07	Mon 14.01.08
19	✓	Final data design	Sat 15.12.07	Wed 26.12.07
20	▤	Final GUI design	Thu 27.12.07	Fri 18.01.08
21	✓	Research	Thu 11.10.07	Sun 18.11.07
22	✓	Analyzing AP-400 Datasheet	Thu 18.10.07	Sun 18.11.07
23	✓	Research on LCD	Thu 11.10.07	Tue 30.10.07
24	✓	Looking for LCD displays	Thu 11.10.07	Tue 16.10.07
25	✓	Buying LCD	Wed 17.10.07	Tue 30.10.07

Project: gannt
Date: Fri 18.01.08

Page 1



ID		Task Name	Start	Finish
26		Implementation	Thu 15.11.07	Thu 17.01.08
27		PCB Design	Fri 04.01.08	Tue 15.01.08
28	✓	Learning Altium Designer	Fri 04.01.08	Sun 06.01.08
29	✓	Creating SchDoc	Sun 06.01.08	Thu 10.01.08
30	✓	Creating PCbDoc	Thu 10.01.08	Mon 14.01.08
31	✓	Buying Necessary Compo	Mon 14.01.08	Mon 14.01.08
32	▦	Printing the Design	Tue 15.01.08	Tue 15.01.08
33		Initializing AP-400	Thu 15.11.07	Sun 25.11.07
34	▦	Comm. with AP-400	Thu 15.11.07	Wed 21.11.07
35	✓	Getting MAC address etc.	Thu 22.11.07	Sun 25.11.07
36	✓	Working with PIC 16F877	Mon 26.11.07	Fri 04.01.08
37	✓	Writing on LCD	Mon 26.11.07	Mon 17.12.07
38	✓	Implementing the DEMO	Wed 26.12.07	Thu 17.01.08

Project: gannt
Date: Fri 18.01.08

Task	▨	Milestone	◆	External Tasks	▨
Split	┈	Summary	▬	External Milestone	◆
Progress	▬	Project Summary	▬	Deadline	⇩

Page 2