

HOME AUTOMATION SYSTEM DESIGN USING ROUND ROBIN SCHEDULING ALGORITHM

Kapil Budhathoki, Professor Dr. M. Asha Rani

Jawaharlal Nehru Technological University Hyderabad, India

Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad, India

Abstract – The high-tech and most sophisticated facilities ever that we have been enjoying in our life have only been possible due to the conscientious research and development over a long span of time. The main focus of this project is in the design of a home automation system or a smart home providing new and exciting facility within. Based on the scheduling algorithm, status of our home is monitored through the use of various sensors. To increase the features, controlling of appliances through voice command has been incorporated. Thus this project adds flexibility addressing the potential problems faced by physically-challenged or handicapped type of people. [1]

Key Words: Home Automation, IoT, Round Robin, ARM7, LPC2148, Speech Processing

1. INTRODUCTION

Home automation or smart home has always been an interest for us. In such field also the technology is gradually becoming matured while providing high end security along with smart features. Providing real-time surveillance is a major concern for modern security systems.

Furthermore rapidly developing technology does not seem to fully confront the problems of physically-challenged or handicapped type of people. The report prepared jointly by the World Health Organization and the World Bank in June 2011 shows that about 15 percent of the world's population, which accounts 785 million people, has significant physical or mental disability.

Thus this project has addressed some of those cases. By the use of voice recognition module different home appliances are controlled. This helps such people to control home appliances according to their voice commands. Furthermore various sensors like PIR sensor, smoke sensor and temperature sensor are used to detect intruder, smoke and temperature. Alert messages are then sent to some pre-defined cell of the family members. [2]

This project makes use of ARM7 based LPC2148 microcontroller and PIC microcontroller to interface all the sensors and voice module.

2. PROJECT BLOCK DIAGRAM

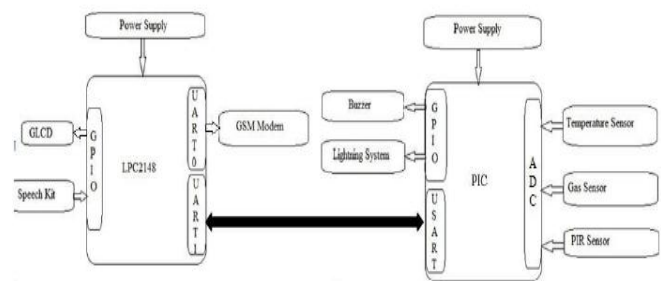


Figure 1: Project Block Diagram

The overall block diagram of this project is as shown in figure 1 which consists of two intelligent units, i.e. two controllers. One is a main control unit which is developed on a LPC2148 board while the other unit, controlled through PIC microcontroller, is interfaced with three different sensors to monitor the status of home.

During the hazardous situation at home, a text message is sent to the cell phone of family members through GSM modem. Also home automation has been added by interfacing a speech kit such that home appliances can be controlled according to the given voice commands. The main unit consists of GLCD to display the different parameters of home status. Here three sensors have been used that will be continually monitoring the home status. The sensors used in this project are temperature sensor, gas sensor and PIR sensor. [7]

The temperature sensor (LM35) has been used to measure the surrounding temperature. It is connected to the port PA.0 of PIC microcontroller as shown in the figure 2. Its output voltage value is proportional to the temperature in °C. It can measure temperature in the range of -55°C to 150°C. The voltage signal that is generated by the temperature sensor is analog in nature. The conversion of this analog signal into its corresponding digital value is

based on successive approximation method. The analog to digital conversion module is built in feature of PIC microcontroller. Each analog sample takes about 2.44 μs to convert into its corresponding digital value. The PIC16F877A microcontroller by default has 10-bit resolution, i.e. the conversion of an analog signal results in corresponding 10-bit digital number. [8]

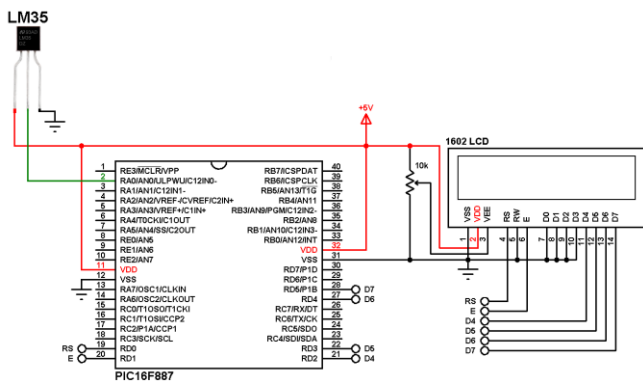


Figure 2: Connecting LM35 to PIC Controller

The resolution is given by the formula

$$\text{Resolution of ADC} = \frac{(+V_{ref} - (-V_{ref}))}{(2^{10}-1)} \quad (2.1)$$

where V_{ref} is reference voltage.

This gives the minimum voltage required to change the ADC result by one bit. Thus there are 1023 voltage steps between 0 volts and +5 volts. So each step is about 5000mV/1023 steps. Suppose the digital value of temperature sensor of 45 means $45 * (5000/1023) = 220mV$.

Since $10mV = 10C$ thus it corresponds to 220C. In this way we can get the temperature value in degree centigrade scale. After converting the analog voltage signal into digital form it is transmitted through the USART port of PIC microcontroller to the LPC2148 board for displaying the temperature value in GLCD. The data transmission is done through RS 232 level serial cable. The pin configuration for 9 pin connector on a DTE device is shown below.

Table 1: Pin Connector in DB9 Cable

Pin No.	Direction of Signal
1.	Carrier Detect (CD) (from DCE) incoming signal from a modem
2.	Received Data (RD) Incoming Data from a DCE
3.	Transmitted Data (TD) Outgoing Data to a DCE
4.	Data Terminal Ready (DTR) Outgoing handshaking signal

5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

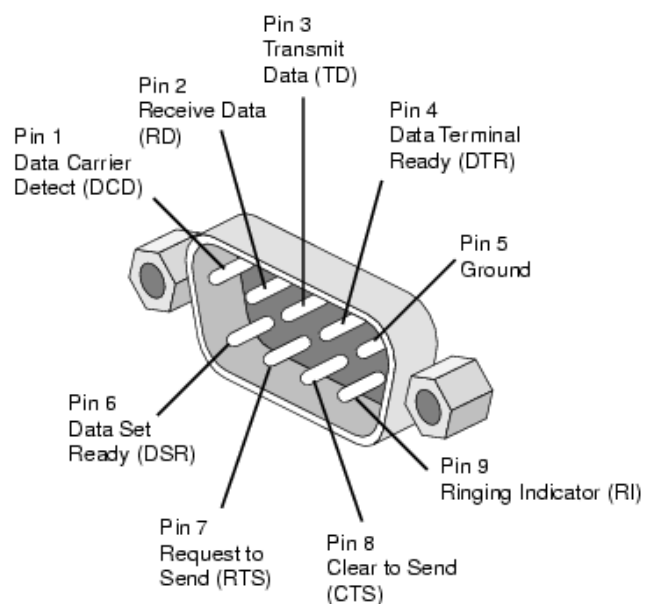


Figure 3: Male RS232 DB9

Here we connect only second, third and fifth pins for serial communication through UART ports between PIC and LPC2148 microcontrollers. Through TD wire data from a DTE device is transmitted to a DCE device. This wire is used by a DCE device to receive its data. The RD wire is the one on which data is received by a DTE device. The connection of UART port of LPC2148 microcontroller with DB9 is as shown in the figure 4. It is connected through MAX3232 which converts signals from RS-232 serial port to signals suitable for use in TTL compatible logic. When a MAX3232 IC receives a TTL level to convert, it changes TTL logic 0 to between +3 and +15 V, and changes TTL logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. The configuration of the registers use for UART in LPC2148 and PIC microcontroller has been discussed in the later section. Once the digital data get transmitted to the LPC2148 microcontroller, it is converted into the corresponding temperature value in centigrade scale. Finally the temperature is displayed in

appliances like light bulb. The light bulb can be turned ON or OFF based on your voice commands. So it increases usability of the system as disabled or handicapped people can also use it at ease.

The main unit of the speech kit is HM2007 speech recognition circuit. It also provides flexibility because it can be trained in any languages. For the speech kit to operate, it must be first trained properly.

Speech is typically sampled at a high frequency, e.g. 16 KHz over a microphone or 8 KHz over a telephone. Then we get a discrete sequence of amplitude values over time domain. A speech recognition system involves some steps as explained below:

1. Converting sound into electrical signals: When we speak into microphone it converts sound waves into electrical signal. The electrical signal strength from the microphone varies in amplitude overtime and is referred to as an analogue signal or an analogue waveform.
2. Background noise removal: The ASR program removes all the noise and gets the words that have been spoken.
3. Breaking up words into phonemes: The words are broken down into individual sounds known as phonemes, which are the smallest sound units noticeable.
4. Matching and choosing character combination: this is the most complex phase. The program has big dictionary of popular words that exist in the language. Each Phoneme is matched against the sounds and converted into appropriate character group. This is where problem begins. It checks and compares words that are similar in sound with what they have heard. All these similar words are collected.
5. Language analysis: here it checks if the language allows a particular syllable to appear after another.
6. After that, there will be grammar check. It tries to find out whether or not the combination of words any sense.
7. Finally the numerous words constitution the speech recognition programs come with their own word processor, some can work with other word processing package like MS word and word perfect.

The main advantage of the IC HM 2007 is that it provides a complete package for speech detection purpose.

2.2 Speech Acquisition

During speech acquisition, speech samples are obtained from the microphone in real time and stored in memory for preprocessing. Speech acquisition requires a

microphone coupled with an analog-to-digital converter (ADC) that has the proper amplification to receive the voice signal, sample it, and convert it into digital speech. The system sends the analog speech through a transducer, amplifies it, and sends it through an ADC. The received samples are stored into memory on a RAM. We can easily implement speech acquisition with the HM2007 IC. The microphone input port with the audio codec receives the signal, amplifies it, and converts it into 8-bit PCM digital samples at a sampling rate of 3.57MHZ. The HM 2007 IC requires initial configuration or training of words, which is performed using a programming board. In the training process user trains this IC by speaking words into the microphone and assigning a particular value for that word. For example a word "ON" can be assigned a value 04 or 18. This can then be later connected to a microcontroller for further functions.

2.3 Speech Preprocessing

The speech signal consists of the uttered digit along with a pause period and background noise. Preprocessing reduces the amount of processing required in later stages. Generally, preprocessing involves taking the speech samples as input, blocking the samples into frames, and returning a unique pattern for each sample, as described in the following steps.

1. The system must identify useful or significant samples from the speech signal. To accomplish this goal, the system divides the speech samples into overlapped frames.
2. The system checks the frames for voice activity using endpoint detection and energy threshold calculations.
3. The speech samples are passed through a pre-emphasis filter.

2.4 Training the IC

An important part of speech recognition kit is training. Training involves creating a pattern representative of the features of a class using one or more test patterns that correspond to speech sounds of the same class. The resulting pattern (generally called a reference pattern) is an example or template, derived from some type of averaging technique. It can also be a model that characterizes the reference pattern statistics.

The keypad and digital display are used to communicate with and program the HM2007 chip. The keypad is made up of 12 normally open momentary contact switches. When the circuit is turned on, "00" is on the digital display, the red LED (READY) is lit and the circuit waits for a command.

2.5 Functional Description

Training or Clearing the pattern:

During training every word is stored to a specific two digit number. So a particular word that has been stored can be cleared. After a word number is entered, CLR function key pressed clears that particular word. All the saved words can be cleared at a time. For this enter 99 through keypad then CLR function key after RESTART. During the clearing process red led will blink continuously.

To train the speech kit first any two digit number should be chosen except 00 and 99. After entering a two digit number, TRN function key is pressed and the required word is spoken. That word is saved to the corresponding word number.

If more than two digits are entered only the last two digits will be taken into count.

For example:

13 CLR clears the pattern corresponding to the word number 13

25 TRN trains the pattern for word number 25

2345 TRN trains the pattern for word number 45

2) Recognition Mode:

a) WAIT pin "L"

In this case, no voice input is accepted until WAIT pin is back to "H" state.

b) WAIT pin "H"

Here RDY is set to low and the speech kit becomes ready to accept the voice input for recognizing. We must train the word patterns before beginning the recognition process. After the completion of recognition process, the result will appear in the D-bus with the DEN pin active.

The speech kit, in this project, has been used to control the home appliances like light bulb in this case. The turning of light ON has been stored to word number 07 while turning of light OFF has been stored to word number 08. The port P1.16 has been used to turn ON the light while port P1.17 is used to turn OFF the light. [9]

2.6 GLCD

The general 16*2 LCDs have their own limitations that they can display only characters and numbers. In Graphical LCDs images can also be displayed. The

Graphical LCDs find applications in mobile screens, video games and other display units.

The GLCD used in my project is CFAG12864B which has a display format of 128*64 dots. The GLCD is divided into two equal sections for internal operations.

Each half is controlled by a controller. The figure 5 explains about the internal structure of 128*64 GLCD. From the figure above we can see that there are 128 columns and 64 rows making a total of 8192 pixels. Each half consists of eight pages. CS in the diagram above stands for controller (chip) select. Each page again consists of eight rows and 64 columns. Thus two horizontal pages make 128 (64*2) columns and eight vertical pages make 64 (8*8) rows.

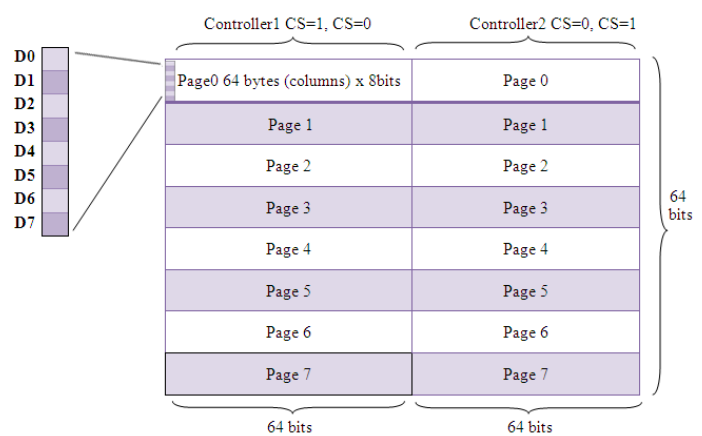


Figure 5: Internal Structure of GLCD

Unlike, general LCD, every dot in the page should be taken into considerations whether to set it high or low while displaying any character, symbol or image. To display those characters, images or symbols we have to choose an appropriate matrix size. The characters can be displayed in the following pattern:

$$A = \{127, 9, 9, 127, 0, 0\}$$

$$B = \{127, 73, 73, 54, 00\} \text{ and so on.}$$

Table 2: Pin description of GLCD

Pin No	Function	Pin Name
1	Ground (0V)	V _{SS}
2	Supply Voltage (5V)	V _{CC}
3	Contrast adjustment	V ₀
4	High for data display/Low for instruction code	Register Select (RS)
5	Low to write to register/High to read from	Read/Write (RW)

	register	
6	Read data when High/Write data at High to Low transition (falling edge)	Enable (EN)
7	Data Pin	DB0
8	Data Pin	DB1
9	Data Pin	DB2
10	Data Pin	DB3
11	Data Pin	DB4
12	Data Pin	DB5
13	Data Pin	DB6
14	Data Pin	DB7
15	Chip selection for IC1; Active high	CS1
16	Chip selection for IC2; Active high	CS2
17	Reset Signal; Active low	RST
18	Output Voltage for LCD driving	V _{OUT}
19	Backlight V _{CC} (5V)	LED
20	Backlight Ground (0V)	LED

		the LCD – set high to read from the LCD to the MCU, set low to write from the MCU to the LCD	
16	D/I	Data/Instruction – tells the LCD whether or not data is being written to the screen or the MCU is using the data bits to perform an instruction – set high for data transfer and set low to designate and instruction is being performed	C.4
17	E	Enable – The enable is used to clock operations to the LCD – it ends up being the clock of the LCD and when it clocks, sequential instructions are performed	C.5

The CFAG12864B has 6 control bits which are used to control the operation and display data on the LCD. They are listed below along with their port assignments.

Table 3: Control bits pin assignment

Pin No.	Symbol	Function/Typical Value	Port Assignment
12	CS1	Column/Chip Select 1- Selects the left driver which is left half of screen	C.0
13	CS2	Column/Chip Select 2- Selects the right driver which is right half of screen	C.1
14	RST	Reset – set high to reset the display, low otherwise	C.2
15	R/W	Read/Write – Selects the type of instruction being performed on	C.3

The most commonly used and also confusing control bits are R/W, D/I, E. Enable must be set low and then high in order for an operation to be passed to the LCD. R/W and D/I are used to determine the mode of operation of the LCD. The following table will illustrate how these control bits are used to control operations.

Table 4: D/I and R/W modes of Operation

D/I	R/W	Mode
0	0	An instruction is being written (such as define Y address)
0	1	MCU reads the status of the LCD – whether it is busy or ready for another command
1	0	Data write – data is written to the display ram at whatever X-Y coordinates have been set
1	1	Data read – the data in the display RAM is read to the MCU

1) Displaying into GLCD

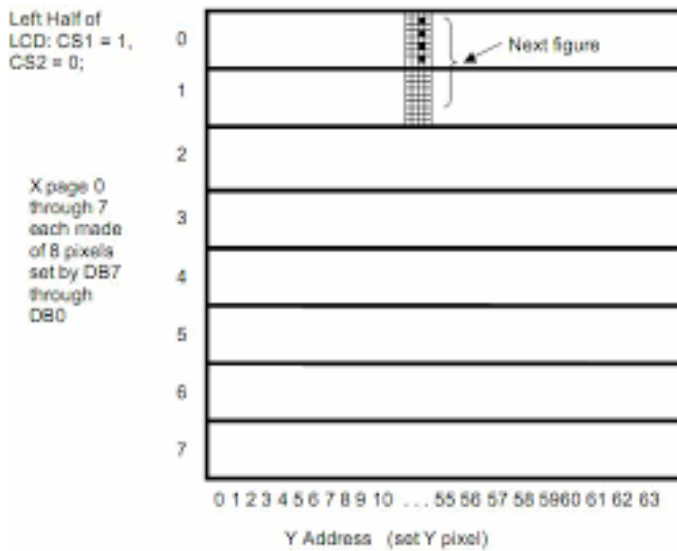


Figure 6: Illustrating Left Half (or Right Half if CS2 = 1 and CS1 = 0)

There are about 8192 pixels on a 128*64 pixel GLCD screen and each pixel is controlled by a series of instructions. Figure 6 shows the left half of the screen (if CS1 is 1 and CS2 is 0). The Y address refers to which line the pixels should be written to and the X page sets the column to which they will be written to.

There are basically three steps that must be done in order to determine where the pixels will go. First the Y address must be set. The Y address actually has a counter so it needs only to be set once and then every time there is a data write it will be incremented to the next line. Secondly the X page should be set. This determines which column of the screen will now be written to. Third, data write command should be issued. Every data bits made high will be darkened on the Y address line in the X page. The driver scans through the pages, using internal Y address counter to its advantage and resetting the X page (and eventually the CS) as it scans through the lines.

An example of displaying a word is as shown below:

```
CHIP_SEL(TWO); // Selecting the second chip controller
GLCD_CMD(0xBC); // SET X PAGE TO 4
GLCD_CMD(0x40); // SET Y COL TO 0
GLCD_str("TEMP="); // Writing Data Command
```

The process of selecting where on the screen the bit is going to be displayed is just a narrowing process. CS1 and CS2 narrow it down to half the screen. Y address narrows it down to a 1 bit wide stripe of height 64 in that half of the screen. X page picks an 8 bit chunk of this 64 bit stripe and

written data writes 8 bits to that chunk. This is the way of putting bits on the screen.

Here GLCD has been used to display the home status like temperature of the house. It also displays the alert message like "SMOKE DETECTED" and "PERSON DEECTED".

The various data pins of GLCD are connected to port P0.16 to port P0.23 respectively of LPC2148 board. The other pins CS1, CS2, RS, RW, EN and RST are connected to port P1.18 to port P1.23 respectively. [11]

2.7 Message Transmission

Global system for mobile communication (GSM) is a standard for mobile communication formed in 1982. A GSM modem works in a wireless radio network. Here GSM modem has been used to send text message to the family member's cell phone while there is any hazardous situation at home. This GSM modem has been connected to the UART0 of LPC2148 board. For the successful transmission of message UART port of LPC2148 must be configured.

Configuring UART in LPC2148

The LPC2148 contains two UARTs. In this project UART0 has been used for connecting the GSM modem. UART uses TxD (Transmit) Pin for transmitting data and RxD (Receive) Pin to receive data. UART sends & receives data in form of chunks or packets which are also referred to as „transmission characters“. The structure of a UART data packet is as shown in figure 7.

LPC2148 has two UART blocks which are UART0 and UART1. For UART0 the TxD pin is P0.0 and RxD pin is P0.1 and similarly for UART1 the TxD pin is P0.8 and RxD pin is P0.9. Both UART0 and UART1 blocks internally have a 16-byte FIFO (First in First Out) structure to hold the Tx and Rx data. Each byte in this FIFO represents a character which was sent or received in order. Both blocks also contain two registers each, for data access and assembly.



Figure 7: UART Packet

The Tx has THR (Transmit Holding Register) and TSR (Transmit Shift Register). When we write data to be sent into THR it is then transferred to TSR which assembles the data to be transmitted via Tx Pin.

Similarly Rx has RSR (Receive Shift Register) and RBR (Receive Buffer Register). When a valid data is received at Rx Pin it is first assembled in RSR and then passed in to Rx FIFO which can be then accessed via RBR.

Registers used for UART of LPC2148

We must first initialize and configure the UART block before those pins can be used for data transmission. Some of the registers used during data transferring are described below.

U0THR (Transmit Holding Register):

It is an eight bit long register used to store the transmission of characters. We must set DLAB (Divisor Latch Access) bit to 0 to access U0THR for write operation.

U0RBR (Receive Buffer Register):

This is an eight bit long register used to store the serially received data. To properly read data from U0RBR, we must set DLAB=0, similar to U0THR.

U0DLL and U0DLM (Divisor Latch Registers):

Both of these eight bit registers are combined together to form a 16-bit divisor value which is used to generate the baud rate. The U0DLL holds the lower 8-bits while U0DLM holds the upper 8-bits making a total of 16 bits with the formation of "[U0DLM:U0DLL]". As they form a divisor value, so the starting value for U0DLL is 0x01, since division by zero is invalid. Thus the starting value in combined formation is 0x0001. In order to access and use these registers properly, DLAB bit in U0LCR must be first set to 1.

U0LCR (Line Control Register):

It is an eight bit register which is used to configure the UART block (i.e. data format used in transmission).

We should properly configure the Line Control Register which is of the following format.

DLAB	Break Control	Parity Select	Parity Enable	Stop Bit Select	Word Length Select
------	---------------	---------------	---------------	-----------------	--------------------

1. Word Length Select [Bit 0 and 1]: It is used to set the length of an individual data chunk. [00] represents the 5 bit data length. While [01], [10] and [11] represent 6, 7 and 8 bit data lengths respectively.

2. Stop Bit Select [Bit 2]: This bit is used to set the number of stop bits to be used during the transmission of character. 0 is used for using 1 stop bit while 1 is used to represent the use of 2 stop bits.

3. Parity Enable [Bit 3]: This bit is used to represent whether parity has been enabled or not. Making this bit 0 means parity generation and checking and has been disabled while 1 represents that parity has been enabled.

4. Parity Select [Bit 4 and 5]: These bits represent the type of parity that has been used. [00] represents the use of Odd-parity, [01] to represent the even parity, [10] to represent the forced 1 parity and [11] to represent the forced 0 parity.

5. Break Control [Bit 6]: This bit will set the break transmission. 0 is used to disable break transmission while 1 is used to enable it.

6. Divisor Latch Access Bit [Bit 7]: This bit is used to set access to the divisor latch. 0 is used to disable access to the divisor latch while 1 is used to enable it. We can access and configure the U0DLL and U0DLM registers only when it is enabled. Generally we first set DLAB to 1, configure Divisor Latch Registers and again disable it by setting 0.

By U0LCR= 0x83;

we can enable DLAB and configure those Divisor Latch registers and after that by

U0LCR= 0x03;

we can disable DLAB.

U0LSR (Line Status Register):

This 8-bit long register is used to know the status of Rx and Tx blocks. This register is of the following format.

RXFE	TEMT	THRE	BI	FE	PE	OE	RDR
------	------	------	----	----	----	----	-----

1. Receiver Data Ready (RDR)-[Bit 0]: 0 means that U0RBR is empty while 1 means U0RBR contains a valid data.

2. Overrun Error (OE)-[Bit 1]: Overrun is the condition when a character is lost since no data is written into FIFO if it's full. 0 means Overrun has not occurred while 1 means Overrun has occurred.

3. Parity Error (PE)-[Bit 2]: When the value of the parity bit in the received character is in wrong state then a parity error occurs. 0 means parity error has not occurred while 1 indicates the occurrence of parity error.

4. Framing Error (FE)-[Bit 3]: The framing error occurs when stop bit of a received character is zero. 0 means framing error has not occurred while 1 means framing error has been occurred.

5. Break Interrupt (BI)-[Bit 4]: 0 means Break Interrupt has not occurred while 1 means it has occurred.

6. Transmit Holding Register Empty (THRE)-[Bit 5]: 0 means U0THR contains valid data while 1 means it is empty.

7. Transmitter Empty (TEMT)-[Bit 6]: 0 means U0THR and/or U0RSR contains valid data while 1 means both U0THR and U0RSR are empty.

8. Error in RX FIFO (RXFE)-[Bit 7]: 0 means U0RBR has no Rx errors while 1 means that U0RBR has at least one error.

UART Baud Rate Generation

In the real world scenario there is a very rare chance to get the actual baud rates as desired. Most times the actual baud rate will drift a little about the desired value. In short, a small amount of error in actual baud rate is generally tolerable in most systems.

The main formula for calculating the baud rate is given as below:

$$\text{BaudRate} = \text{PCLK in hertz} / (16 * (256 * \text{DLM} + \text{DLL}) * (1 + \text{DIV_ADDVAL} / \text{MULVAL}))$$

where PLCK = Peripheral Clock in Hertz

DLM and DLL = Divisor Latch registers

DIDADDVAL and MULVAL = Fractional baud rate generator registers

The simplest method to calculate DLM and DLL is given below:

$$\text{DLL} = (\text{PCLK} / (16 * \text{Desired_BaudRate})) \% 256$$

$$\text{DLM} = (\text{PCLK} / (16 * \text{Desired_BaudRate})) / 256$$

Thus for PCLK = 30 Mhz and for baud rate of 9600 we get

$$\text{DLL} = 195 \text{ and}$$

$$\text{DLM} = 0$$

At first we must set DLAB to 1, thus the line control register is initially configured as

$$\text{U0LCR} = 0x83;$$

Then we set the values for DLL and DLM as follows

$$\text{U0DLM} = 0;$$

$$\text{U0DLL} = 195;$$

In the next step we again configure the LCR defining other parameters as follows

$$\text{U0LCR} = 0x03;$$

The data to be transmitted is first stored in THR. After every data transmitted, LSR is checked to know the completion of data transfer as follows

```
U0THR = data;
```

```
while((U0LSR & 0x40)==0);
```

When all the data has been transmitted then the 7th bit in LSR will be high i.e. 1. In such a case when it is ANDed with 0x40 the result will not be equal to 0. Thus the data transmission function will come out of the loop and stop transmission. Else, the ANDed result equals 0 and the data transmission process continues and until the end.

At the receiver end, the data received is stored in RBR. As the data is received, LSR is checked to know about the completion of data reception as follows

```
while((U0LSR & 0x01)==0);
```

```
return(U0RBR);
```

When all the data reception has been completed, the first bit in the LSR will be high i.e. 1. In such a case, when it is ANDed with 0x01 the result will not be equal to 0. Thus the data reception function will come out of the loop and stop receiving the data. Else, the ANDed result equals 0 and the data reception process will continue until all the data has been received.

An example of GSM AT commands for sending message is as shown below:

```
UART0_SendStr("AT+CMGS=\"+918179150XXX\"r");
```

```
delay(10);
```

```
UART0_SendStr ("ALERT!!! SMOKE DETECTED AT HOME");
```

First the cell number to which message has to be transmitted is initialized using AT commands. After that required string is sent.

Configuring USART of PIC Microcontroller

Most of the Microchip's PIC microcontrollers have built in USART Module. Here we will discuss about configuring the registers for data communication for PIC microcontrollers.

PIC16F877A USART PIC Microcontroller

Transmit Status and Control Register – TXSTA

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Figure 8: Transmit Status and Control Register

1. Clock Source Select Bit (CSRC)-[Bit 7]: This bit has no application in Asynchronous mode operation while used to select master or slave mode in Synchronous mode operation.

2. 9-bit Transmit Enable Bit (TX9)-[Bit 6]: This bit when set enables the 9 bit transmission otherwise 8 bit transmission mode is used. The 9th bit is used as a parity bit.

3. Transmit Enable Bit (TXEN)-[Bit 5]: This bit when set enables the transmission. In the synchronous mode operation CERN and SREN bits of RCSTA register overrides this bit.

4. USART mode select Bit (SYN)-[Bit 4]: This bit is used to select the mode of transmission.

1 = Synchronous mode
0 = Asynchronous mode

5. Unimplemented Bit [Bit 3]: It is not used and read as 0.

6. High Baud Rate Select Bit (BRGH)-[Bit 2]: This bit is used in Asynchronous mode operation and unused in Synchronous mode. Setting this bit enables High Speed while clearing it enables Low Speed baud rates.

7. Transmit Shift Register Status Bit (TRMT)-[Bit 1]: This is Transmit Shift Register (TSR) status bit. It is used to check whether the data written to transmit register is transmitted or not.

1 = TSR empty

0 = TSR full

8. 9th Bit of Transmit Data (TX9D)-[Bit 0]: This is the 9th bit of data in the 9 bit transmission mode. It is commonly used as parity bit.

Receive Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Figure 9: Receive Status and Control Register

1. Serial Port Enable Bit (SPEN)-[Bit 7]: This bit enables the serial port.

1 = Serial port enabled; configures RC7/RX/DT and RC6/TX/CK pins as serial port pins

0 = Serial port disabled

2. 9-Bit Receive Enable bit (RX9)-[Bit 6]: Setting this bit enables 9 bit reception otherwise it will be in 8 bit reception mode.

3. Single Receive Enable Bit (SREN)-[Bit 5]: This bit has no effect on Asynchronous mode and Synchronous Slave mode. Setting this bit enables Single Receive. This bit is cleared after the reception is complete.

4. Continuous Receive Enable Bit (CREN)-[Bit 4]: Setting this bit enables Continuous Receive. In Synchronous mode CREN overrides SREN.

5. Address Detect Enable Bit (ADDEN)-[Bit 3]: This bit is applicable only in Asynchronous 9 bit mode. Setting this bit enables Address Detect.

6. Framing Error Bit (FERR)-[Bit 2]: Detecting 1 at this position stands for Framing Error while 0 means No Framing Error.

7. Overrun Error (OERR)-[Bit 1]: A high value at this position indicates that Overrun Error has occurred.

8. 9th Bit of received Data (RX9D)-[Bit 0]: This is the 9th bit of Received Data and is commonly used as Parity Bit.

USART Baud Rate Generator (BRG)

Baud Rate Generator provides the required clock for the data transmission and reception. USART module has a dedicated 8-bit baud rate generator that supports both Synchronous and Asynchronous modes. The 8-bit SPBRG register controls the time period of the free running timer. Baud rate can be calculated from the following equations; where FOSC is the clock frequency of the microcontroller.

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = FOSC/(64 (X + 1))	Baud Rate = FOSC/(16 (X + 1))
1	(Synchronous) Baud Rate = FOSC/(4 (X + 1))	N/A

Legend: X=value of SPBRG (0 to 255)

Figure 10: PIC USART Baud Rate Formula

7) PIC ADC Registers

ADCON0 Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

Figure 11: ADCON0 Register

1. ADCS1 and ADCS2: These two bits are used to select A/D Conversion Clock. It should be selected according to the device clock.

2. CH2, CH1 and CH0: They are used to select one out of eight analog input channels.

3. GO/DONE: It gives the A/D Conversion Status. Setting this bit initializes A/D Conversion and will be automatically cleared when the conversion is complete.

4. ADON: This bit is used to switch ON/OFF the ADC module. Setting 1 turns ON while 0 turns OFF the module.

ADCON1 Register

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Figure 12: ADCON1 Register

1. ADFM: It is an ADC Result Format select bit. Two 8 bit registers (ADRESH and ADRESL) are provided to store the 10-bit result of A/D Conversion. When ADFM is 1, the result will be right justified, i.e. Most Significant Bits of ADRESH will be read as 0. When ADFM is 0, the result will be left justified, i.e. Least Significant Bits of ADRESL will be read as zero.

2. ADCS2: It is used to select A/D Conversion Clock in association with ADCS1 and ADCS2 of ADCON0 register.

3. PCFG3-PCFG0: These are the A/D Port Configuration Control bits. Each pin among AN0 – AN7 is configured as analog, digital or reference voltage inputs according to the status of these configuration bits as given below.

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

The sensors have been connected to the PORTA of PIC microcontroller. So the PORTA must be configured as input port. This is done as follows:

TRISA = 0xFF;

Then the A/D Conversion Clock and analog input channels are configured as

ADCON0 = 0x41;

This sets the A/D Conversion Clock to Fosc/8 and also selects the Analog Channel 0. Finally the ADON set to 1 turns ON the converter module.

Now the format of storing the A/D result and A/D Port Configuration Control bits is set as follows:

ADCON1= 0x02;

This sets the A/D Result Format to left justified such that the Least Significant Bits of ADRESL will be read as zero. Next three analog input channels are selected.

Now we configure the Transmit Status and Control Register as follows:

TXSTA = 0x24;

This sets 8-bit asynchronous mode of transmission. Also transmit is enabled with High Baud rate. The Transmit Shift Register Status bit sets TSR full. The Receive Status and Control Register is configured as follows:

RCSTA = 0x90;

This configuration enables serial port and sets 8-bit reception. It also sets continuous reception. Here no framing error, no overrun error have been set. [4]

2.8 Successive Approximation Technique

It is a technique used by the A/D conversion blocks for the process of analog to digital conversion in LPC2148 board. This technique performs binary search through all possible quantization levels before finally converging upon a digital output for each conversion. The successive approximation technique for analog to digital conversion has four major components:

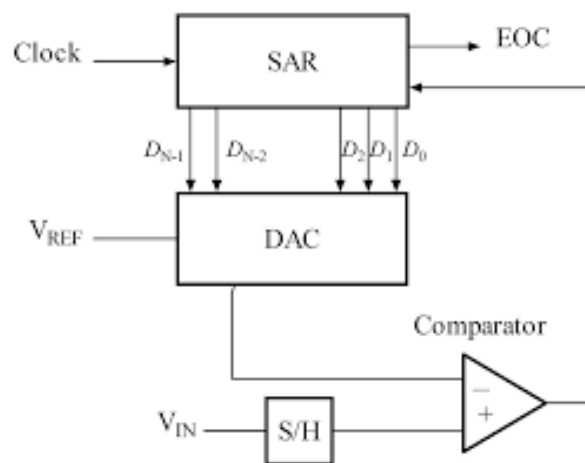


Figure 13: Successive Approximation ADC Block Diagram

1. A sample and hold circuit to hold the input voltage, V_{in}
2. An analog voltage comparator that compares V_{in} with the output of the internal DAC and feeds the result of comparison to the successive approximation register (SAR).
3. A successive approximation circuit provides an appropriate digital code of V_{in} to the internal DAC.
4. A DAC supplies the comparator with an analog voltage equal to the digital code output of the SAR.

Initially the successive approximation register is initialized in such a way that the most significant bit (MSB) is made equal to 1. Then this code is fed into the DAC which then generates the analog equivalent of this digital code. It is then compared with the sampled input voltage. If this analog voltage exceeds V_{in} , the comparator causes SAR to reset this bit otherwise the bit is left as 1. In the similar way, the next bit is set to 1 and binary search is continued until every bit in the SAR has been tested. The resulting code will be the digital approximate value of the sampled input voltage and is finally output by the SAR at the end of conversion (EOC). [6]

3. Result and Discussion

In this project all the objectives have been successfully achieved. We are able to control the home appliances through voice commands. The speech kit recognizes the given voice commands and controls the home appliances accordingly. The temperature is measured by a temperature sensor LM 35 and its value is displayed in the GLCD. The gas sensor is used to detect the presence of LPG gas leakage or smoke within the house. If it detects gas leakage or smoke within the home, a buzzer will turn ON to notify the situation and also a text message is sent to the cell phone of family member. The GSM modem is used to send the text message. The GSM modem has been interfaced with the LPC2148 board.

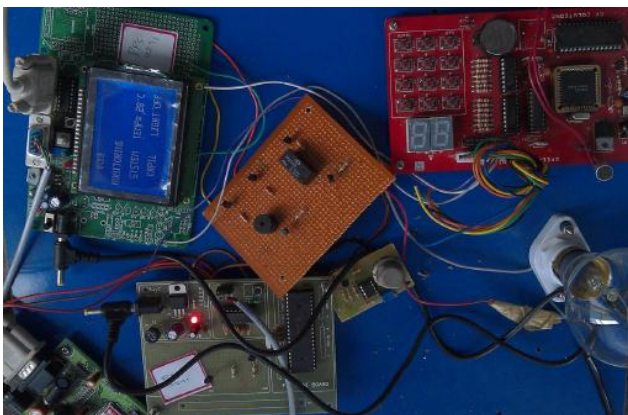


Figure 14: Light OFF Condition

The figure 14, as shown above, demonstrates the light OFF situation of my project. It was turned OFF according to the given voice command. All the modules have also been shown here. It is also showing the value of temperature as "TEMP=280C" in the GLCD. The "LIGHT OFF" situation is also displayed in the GLCD.

In the figure 15 we can see the light has been turned ON. It was turned ON by the voice command given through speech kit. The light bulb has been connected through relay as it needs higher voltage level i.e. 220V.

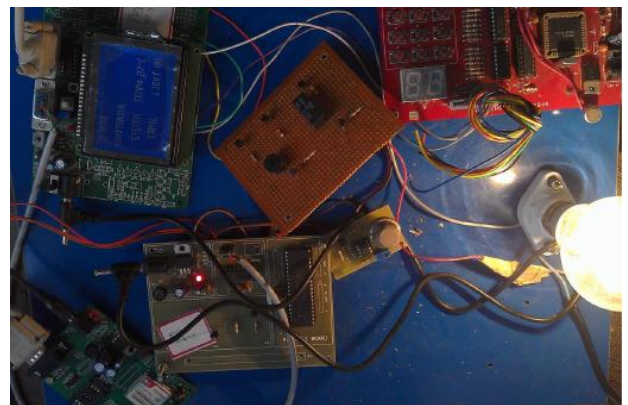


Figure 15: Light ON Condition

The figure 16 shows the message that is received at the cell phone when some gas leakage or smoke is detected at home. As soon as the gas sensor detects hazardous situation at home, an alert message is immediately sent to the cell phone of family members through GSM modem.

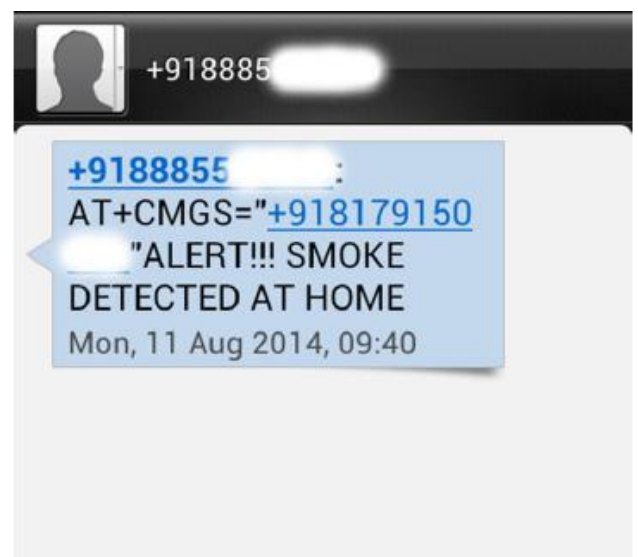


Figure 16: Message received after smoke detected by gas sensor

In the figure 17 we can see that “PERSON DETECTED” has been displayed in the GLCD. As a PIR sensor detects some movement around its peripheral, it will sense the change in IR radiation. Then the message is displayed in the GLD about the situation.

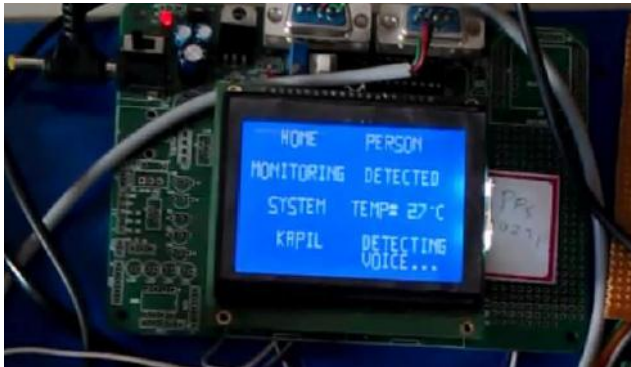


Figure 17: Showing person detected message in GLCD

The figure 18 shows the led glowing when temperature reaches certain value. Here temperature limit has been adjusted to 45°C. So as temperature reaches the preset value led will glow to indicate the temperature exceed.

From the figure 19 as shown above, we can see the alert message being displayed in the GLCD after gas leakage was detected by the sensor. The gas sensor was brought in contact with the lighter that contained butane gas.

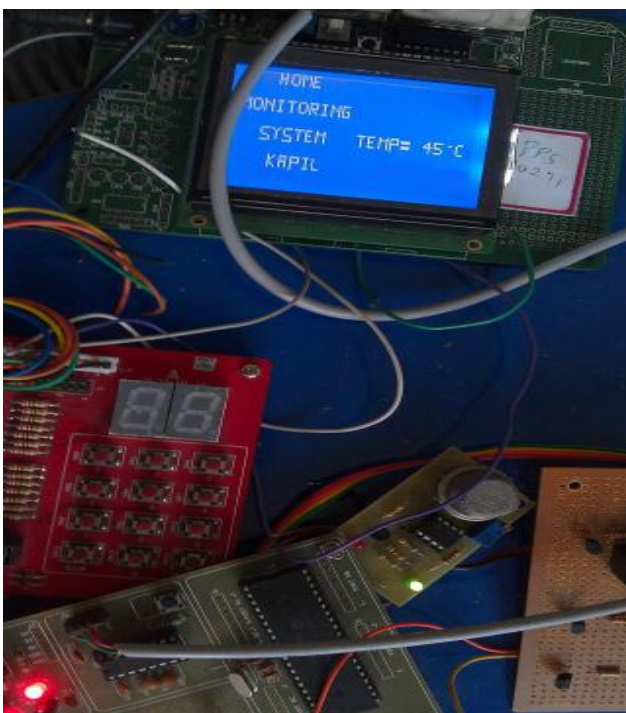


Figure 18: Led glowing when temperature exceeds the limiting value

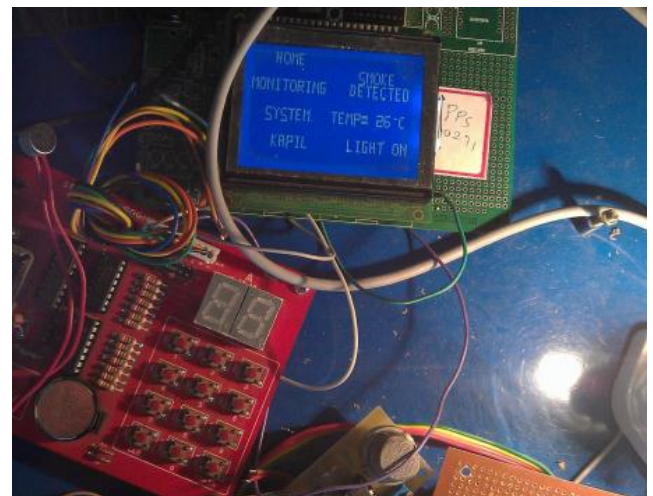


Figure 19: Figure to show when smoke detected

4. LIMITATIONS AND FUTURE ENHANCEMENTS

The speech kit used is not capable of recognizing long phrases. So it should be trained with short words only. Furthermore in the part of home automation only turning light ON and OFF has been taken into considerations. Other features for home automation have not been included. The sensors have been used to monitor the home status in round-robin fashion. Other methods can be implemented.

In the case of future enhancement, high quality speech kit can be used so that it can recognize long phrases also. In case of providing home automation, some more features required in the home can be included. Also using RTOS other suitable scheduling algorithms can be incorporated.

5. REFERENCES

- [1] Bo Cui and Tongze Xue, Design and Realization of an Intelligent Access Control System Based on Voice Recognition, 2009 ISECS International Colloquium on Computing, Communication, Control, and Management
- [2] Rakesh V S, Sreesh P R and Sudhish N George, An Improved Real-Time Surveillance System for Home Security System using BeagleBoard SBC, Zigbee and FTP Webserver
- [3] Andrew N. SLOSS, Dominic SYMES and Chris WRIGHT “ARM System Developer“s Guide Design and Optimizing System Software”
- [4] LPC214x User Manual, www.nxp.com/
- [5] <http://en.wikipedia.org/wiki/ARM7>
- [6] Successive approximation ADC,

http://en.wikipedia.org/wiki/Successive_approximation_ADC

[7] Passive Infrared Sensor,
http://en.wikipedia.org/wiki/Passive_infrared_sensor

[8] LM35 Precision temperature measurement and calibration,
<http://www.magicsnotebook.com/lm35.html>

[9] Speech recognition,
http://en.wikipedia.org/wiki/Speech_recognition

[10] AT Commands Interface Guide, 05th April 2002,
WAVECOM

[11] LCD-128H064A,
<http://www.vishay.com/displays/list/product-37329/>