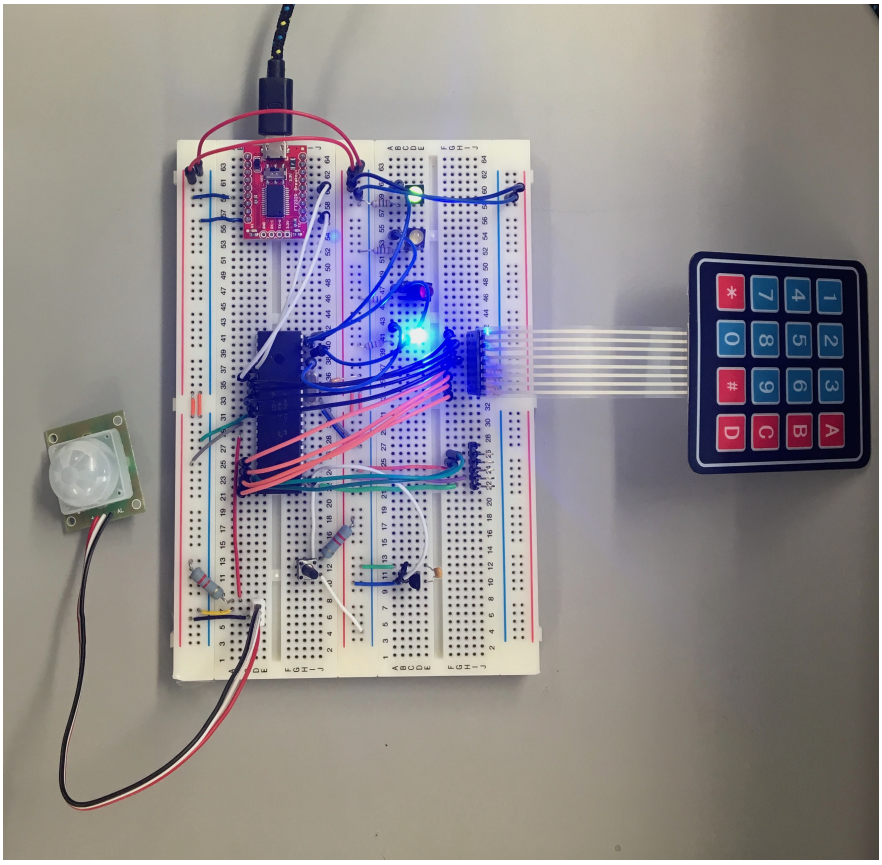
Lab 7

PIC18F4520 Standalone Alarm System with EUSART Communication



Don Kuruppu
1001 – 101 – 220

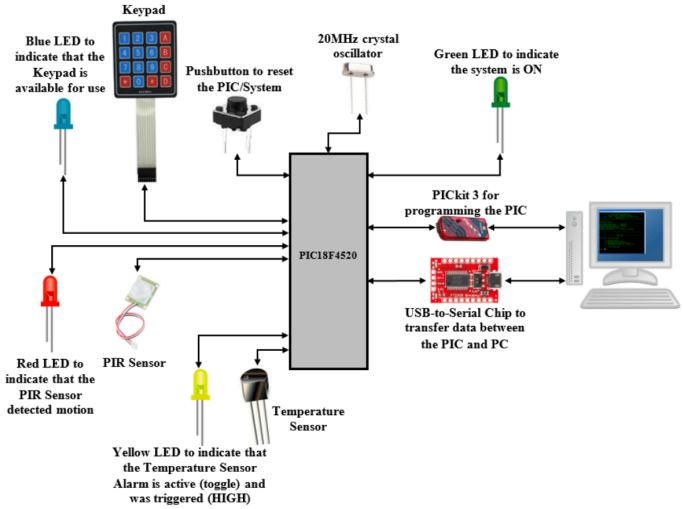
Contents

1. Alarm System Description

This password-protected alarm system is built around a PIC18F4520 microcontroller. The microcontroller is driven by a 20MHz crystal. Communication between the alarm and the user is performed through a serial connection, that uses the USART protocol, that is between the alarm system and a computer. The serial connection is handled by FT232RL chip on the alarm. This chip is the source of power for the alarm as well. Additionally, the alarm system utilizes the EEPROM of the microcontroller to save its state so that it could be restored upon being reset.

The alarm contains a PIR motion sensor to detect motion. The user can enable it to detect motion and whenever it detects motion, it notifies the user. Then, the user must enter the password to clear the notification. Also, it gives the user the option to turn off the motion sensor or leave it on.

Furthermore, the alarm consists of a temperature sensor that drives the ADC module of the microcontroller. The user has the option to turn on/off the temperature sensor. When the user enables the temperature sensor, they must give a threshold temperature. The alarm system, compares the temperature values being read against the threshold temperature and issues a notification whenever a temperature reading exceeds the threshold temperature. Then the user must enter the password to clear the notification and decide whether to leave the temperature sensor on or turn it off.



PIC Alarm System Overview

The alarm system is user-friendly in that it gives the user two methods of input but only method input is active at any given time. The user interfaces into the alarm with a terminal window open on a computer. The user then can send input through the keyboard or the matrix pad attached to the alarm.

The alarm system also provides the user the ability to reset it in the form of a push button. Additionally, it contains four LEDs to notify the user about the state of the alarm. The green LED when illuminated notifies the system being powered on. The yellow LED blinks when a new temperature reading is being processed and stays solid whenever the temperature sensor issues a notification. The red LED illuminates whenever the alarm detects motion. As for the blue LED, it shows the user which method of input is active.

2. Textual Overview of Planning and Execution of the Project

The plan to complete the project was straightforward. The first step was to read the documents given to understand the requirements of the alarm. Afterwards, it was necessary to go meet the instructor of the course to clarify the doubts arose.

After understanding the requirements, the next step was to integrate the hardware components. Firstly, the microcontroller was setup and powered. Next, the 20MHz oscillator was installed and an LED was made to blink with a known frequency to establish that the oscillator was working properly. Afterwards, the serial communication was established and tested. Next, installed the master clear switch.

Once the fundamental hardware items were working, started implementing the matrix pad. Next, got the PIR sensor and temperature sensor operational.

When all the components were operational individually, the next task was to integrate them and get them function as a unit. Afterwards, installed the LEDs and got them to illuminate at the right times.

Once setting up hardware was finished, started on designing the alarm interface with the user and implemented alarm functionalities got the alarm system operational.

Afterwards, started testing the functionalities of the alarm system and tested every possible scenario corrected the defects found and finalized everything.

3. Description of the PIC Interrupts Used

3.1. Motion Sensor Interrupt

PIR Sensor

Motion sensor, which is an interrupt source for the alarm, has the highest priority when compared to all the other sources of interrupts. Therefore, pin zero of Port B, which is only high priority by default was selected to connect the PIR motion sensor to the microcontroller. The motion sensor stays high and would drop low whenever it detects motion. Therefore, pin zero of Port B was adjusted to trigger an interrupt whenever it detects a falling edge from the motion sensor.

3.2. Temperature Sensor Interrupt

Temperature Sensor

The temperature sensor is connected to the microcontroller's through pin zero of Port A, which is also analog channel zero for the ADC module. It utilizes multiple interrupts. First off, it uses the ADC module of the microcontroller to feed its output voltage to be converted to a sensible temperature value. The ADC module is interrupt driven and is of low priority.

Secondly, it uses timer zero to maintain a two second time the delay between two successive ADC sample acquisitions. Timer zero is also interrupt driven and is of low priority.

4. Calculations

4.1 ADC Calculations

Once you convert the ADC result in ADRESL and ADRESH into a voltage, assume that voltage is named current temp.

Voltage difference = currentTemp - 0.5

The result above is the output voltage difference between 0 Celsius and current temperature in Celsius

Temperature difference = Voltage difference * 100

Calculating how many 0.1mV are there in the voltage difference to get the temperature difference.

Current temperature = 0 + Temperature difference

The temperature difference was between 0 Celsius and current temperature in Celsius

Current Temperature in Fahrenheit = (Current temperature * (9 / 5)) + 32

4.2 Timer Zero Calculations

The purpose of using timer zero was to implement a two second sample delay and it was driven by the instruction frequency.

Number of Ticks = Delay * (FOSC / (4 * pre - scaler))

Number of Ticks = 2 * (20000000 / (4 * 256))

= 39062.5

Preload value = 65536 – 39062

= 26473

= 0X6769

4.2 USART Calculations

This alarm system uses a low speed BAUD rate of 9600.

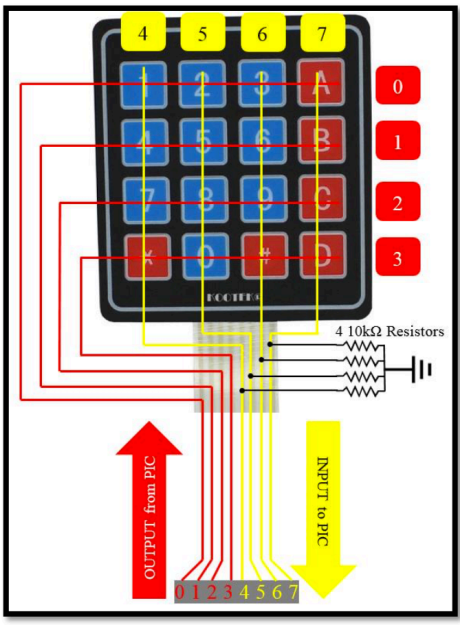
BAUD Rate = FOSC / (4 *16(SPBRG + 1))

9600 = 20000000 / (64(SPBRG + 1))

SPBRG = ((20000000) / (64 * 9600)) – 1

= 12

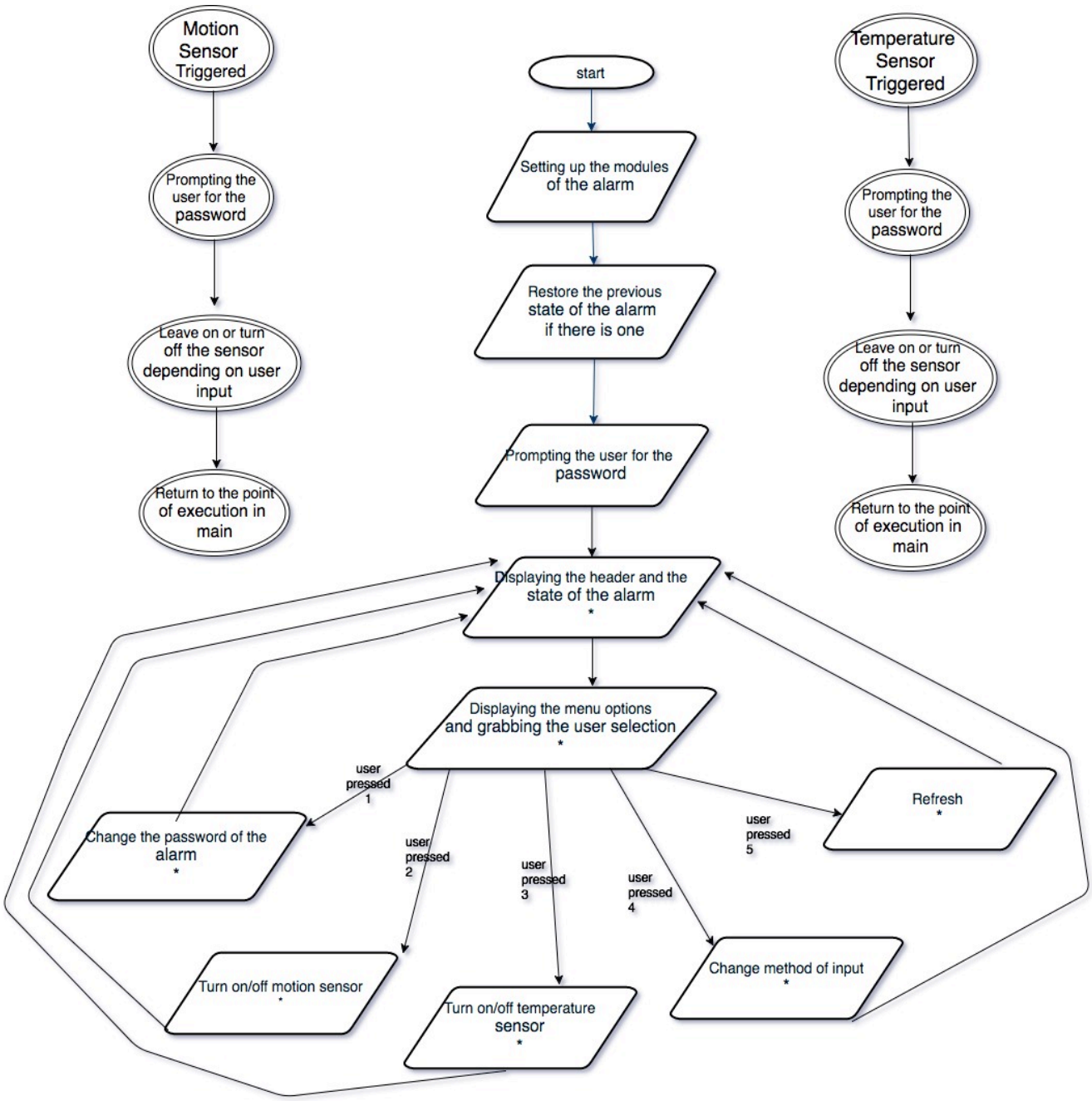
5. Description of the Process of the Matrix Pad



Lines zero through three are powered by the microcontroller. Starting from line zero, lines are driven high one at a time. Therefore, the matrix pad is read in rows. Whenever, the user presses a button it will complete the connection between the corresponding output line and input line and this will send the microcontroller a pulse. Based on what output line was high and which input line the pulse came in through, the key that was pressed could be identified.

Driving the lines high, one at a time, happens so fast that it is fair to assume that at any given key press, the corresponding output line will be high. Therefore, the press would complete the circuit and send the pulse to the microcontroller with the correct input line.

6. Alarm System Flow Chart



**Execution can jump to any of the two interrupts from any state that has an asterisk in it.

7. Problems Encountered

Overall, the execution of the project was straightforward. The main problem that was encountered was running out of program space in the microcontroller to put the hex file in. The instructor pointed out using printf() calls to print double variables generates a significant amount of code that floods the program space making it impossible to have enough free space for the other parts of the program. Fixed the program by removing the printf() calls that printed doubles.

Another problem encountered was the keyboard not accepting input from the user after displaying the main menu, whenever the program returns to the point of waiting for user input from the keyboard after handling an interrupt. To avoid this, the instructor suggested implementing software resets at the end of the motion and temperature sensor interrupts, which solved the problem.

8. Conclusion

The alarm system was designed and implemented successfully while completing all the requirements listed in the lab 7 description document. A suggestion to make this project even more interesting and better is to incorporate another alarm function that utilizes one of the CCP modules of the microcontroller.