HOME AUTOMATION SYSTEM

A Graduate Project Report submitted to Manipal University in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
In
Instrumentation and Control Engineering

Submitted by

Udayan Prabir Sinha
Pranay Raj Singh

Under the guidance of

Mr. Anjan Gudigar

Assistant Professor (Sr. Scale)
Department of Instrumentation and Control Engineering

DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING
MANIPAL INSTITUTE OF TECHNOLOGY
(A Constituent College of Manipal University)
MANIPAL – 576104, KARNATAKA, INDIA

May 2015

DEPARTMENT OF INSTRUMENTATON AND CONTROL ENGINEERING

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent College of Manipal University)

MANIPAL – 576 104 (KARNATAKA), INDIA

Manipal

11th May, 2015

CERTIFICATE

This is to certify that the project titled **HOME AUTOMATON SYSTEM** is a record of the bonafide work done by **UDAYAN PRABIR SINHA** (*Reg. No. 110921560*) and **PRANAY RAJ SINGH** (*Reg. No. 110921278*) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **INSTRUMENTATION AND CONTROL ENGINEERING** of Manipal Institute of Technology Manipal, Karnataka, (A Constituent College of Manipal University), during the academic year 2014-15.

Mr. Anjan Gudigar
Assistant Prof. (Sr. Scale), I & CE
M.I.T, MANIPAL

Prof. Dr. Dayananda Nayak
HOD, I & CE.
M.I.T, MANIPAL

ACKNOWLEDGMENT

We would like to take this opportunity to thank everyone who directly/indirectly helped us with this project.

We owe our deepest gratitude to my guide, Mr. Anjan Gudigar (Assistant Professor, Sr. Scale) whose encouragement, guidance, expertise and support from the start to the finish enabled us to develop an understanding of the field and finally implement this project.

We are indebted to many of our classmates to support us all the way with the project. We would like to show our gratitude to all the lab assistants and faculty members of the Department of Instrumentation & Control Engineering for providing their valuable experience in our time of need.

Lastly, we offer our regards and blessings to all of those who supported us in any respect during the completion of the project.

ABSTRACT

The IoT (Internet of Things) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. With major technology giants touting IoT as the next technological revolution in embedded computing, it must be realized that an efficient and reliable home automation system will serve as the backbone of the IoT-enabled world in the consumer domain.

A prototype of a basic and functional home automation system which supports IoT integration is designed and implemented. Due to the modular nature of the design, the whole system was easily divided into small modules, each of which could be designed, tested and debugged independently before integrating them into one complete, fully-functional system. The system was divided broadly in 4 parts as explained in chapter-3: Design of the MCU-based modules, design of the remote controller, design of the central module and final integration and test-run.

A number of glitches/bugs/shortcomings related to power consumption, faulty functionality and some inherent shortcomings of the platform being used for implementation were observed in action during the final integration and testing. The reasons for the occurrence of those glitches and the corrective measures taken for rectifying them have been described.

In the end, a functional home automation system is successfully designed and implemented. Various areas of future improvement in the design are pointed out in chapter-5. A variety of hardware components (PIC18F4550 MCU, Arduino, Raspberry Pi, nRF24L01+ 2.4GHz RF communications module) and software packages (MPLAB-X IDE for PIC MCUs, Python & associated libraries, Arduino IDE, EAGLE for creating schematics, Lucid Chart for creating flowcharts) were used to design the system and the supporting documentation.

LIST OF TABLES

LIST OF FIGURES

CONTENTS

CHAPTER 1
INTRODUCTION

The IoT (Internet of Things) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. Typically IoT is expected to offer advanced connectivity of devices, systems and services that go beyond standard M2M (machine-to-machine) communications [1]. It is expected to usher in automation in nearly all fields and allow for advanced applications like Smart Grid [1]. A home automation system has been envisioned to be the central hub of IoT-enabled devices in the consumer domain. Hence, it is necessary that a reliable, efficient and smart home automation system be designed to allow for easy integration of such devices.

1.1 NEED FOR THE PROJECT

With major technology giants touting IoT as the next technological revolution in embedded computing, it must be realized that an efficient and reliable home automation system will serve as the backbone of the IoT-enabled world in the consumer domain [1]. A home automation system in an IoT-enabled world will be connected to for example:

- Medical devices such as blood glucose monitors and pace makers which would send regular reports to the doctor via the Internet connection of the home automation system. This will allow for potentially early diagnosis of diseases and better health monitoring [3].

- Automobiles, the on-board car computer will monitor data from various sensors & alert the end-user in situations like when maintenance is required or tire pressure is low or it needs to be refueled.

- Home security systems to automatically alert the emergency services in situations like fire, break-in, etc. [2].

- IoT-enabled home appliances; for example a coffee-maker which automatically brews up coffee for the end-user in the morning when he wakes up & in the evening when he is about to return from work or a smart refrigerator which uses computer vision to monitor food health & alert the user of rotting food [2].

- Wearable computers like smart watches.

- Smart energy meters to intelligently regulate power to the house & ensure that monthly power consumption does not cross a fixed threshold [2].

The nature of IoT warrants a modular and (depending on the application) possibly decentralized design of the system.

1.2 OBJECTIVE

To design and implement a basic and functional home automation system which allows easy integration of IoT-enabled devices.

1.3 PROJECT SCHEDULE

- **January 2015**
  - Design of the system and finalization of its specifications.
  - Selection of components.
- **February 2015**
  - Design of microcontroller based modules.
- **March 2015**
  - Design of the remote controller.
  - Design of the central module.
- **April 2015**
  - Integration of the module into 1 complete system.
  - Complete system test-run and optimization.
- **May 2015**
  - Documentation.

1.4 ORGANIZATION OF REPORT

- **Chapter 1:** Brief introduction of the project and motivation. Project objectives and schedule are also included
- **Chapter 2:** Survey of relevant literature and conclusions drawn from it.
- **Chapter 3:** Project methodology, program flowcharts and working
- **Chapter 4:** Result analysis. Analysis of shortcomings detected and corrective measures taken.
- **Chapter 5:** Future scope of the project discussed.

CHAPTER 2
BACKGROUND THEORY

Mentioned below are some of the relevant literature which that were reviewed and the topics which were found to be relevant to the project in that literature:

- Richard Harper [2], talks about the predictions of leading electronics and consumer companies such as, in the future we may never again have to worry about forgetting to switch the lights off, or locking the front door when we leave our home. Even mundane chores such as visiting the local supermarket to stock up your fridge will become a thing of the past, as the Smart Homes of the future will be able to decide what is needed and order them for us. This book looks at the designs and technologies behind these new innovations, along with their time-saving, environmental and security benefits, amongst others. Despite these clear advantages however, Smart Homes have so far failed to reach the levels of success originally anticipated, thereby forcing designers to further examine the roles and practicalities of these new technologies. Through the use of detailed case studies from such international giants as Orange, Ariston, Philips Electronics and Electrolux, this book clearly demonstrates what will be possible for our domestic settings in the future if we continue to develop the concept of the Smart Home.

- William Mann [3], talks about recent research and technological developments from engineering, computer science, and the rehabilitation sciences, detailing how its applications can promote continuing independence for older persons and those with disabilities. The author has used detailed product descriptions, photographs and illustrations, and case studies, to discuss the influence of cutting-edge technologies on wearable systems, human-computer interactions, assisted vision and hearing, smart wheelchairs, handheld devices and smart phones, visual sensors, home automation, assistive robotic sand in-room monitoring systems. The author examines recent trends in other critical areas, such as basic assistive technologies, driving, transportation and community mobility, home modifications and design, and changing standards of elder care.

- Recently, it is observed that how the Internet of Things (IoT) will deliver a smarter grid to enable more information and connectivity throughout the infrastructure and to homes [1]. The author explains how the IoT, consumers, manufacturers and utility providers will uncover new ways to manage devices and ultimately conserve resources and save money by using smart meters, home gateways, smart plugs and connected appliances. The author discusses, the different approaches being taken worldwide to connect the smart grid. Examples are provided on how Texas Instruments is developing full system solutions by combining hardware (analog and digital) and software to address some of the challenges in building a smarter and more connected smart grid. The author also tells us about the key defining characteristics of an IoT device and the role of a full-fledged home automation system in consumer IoT.

All of the above mentioned literature put forward some important design aspects which must be considered while designing IoT applications:

- Power efficient design.
- Modular and decentralized design.
- Networked design.
- An IoT device does not need to be connected to the Internet directly.

CHAPTER 3
METHODOLOGY

The modular nature of the design allows easy division of the whole system into small modules, each of which can be designed, tested and debugged independently before integrating them into one complete, fully-functional system.

Hence, project design is carried out in the following parts:

- **Design of the microcontroller based modules:**
  - Design of the display module (displays useful information to the user).
  - Design of the relay module (use to control flow of power to appliances, lights, fans, etc.).
  - Design of the door locking module (used to lock/unlock front door of the house).
- **Design of the remote controller** (used by the user to control the system):
  - Selection of interface (preferably touch-screen) & required hardware.
  - Design of software for the interface and transmit input data in a desirable format wirelessly.
- **Design of the central module:**
  - Implement program to fetch weather & time data using an appropriate developer API.
  - Use available libraries to process the data and transmit it in desirable format wirelessly.
- **Final integration and testing:**
  - Optimize and integrate the modules into a complete and functional home automation system.

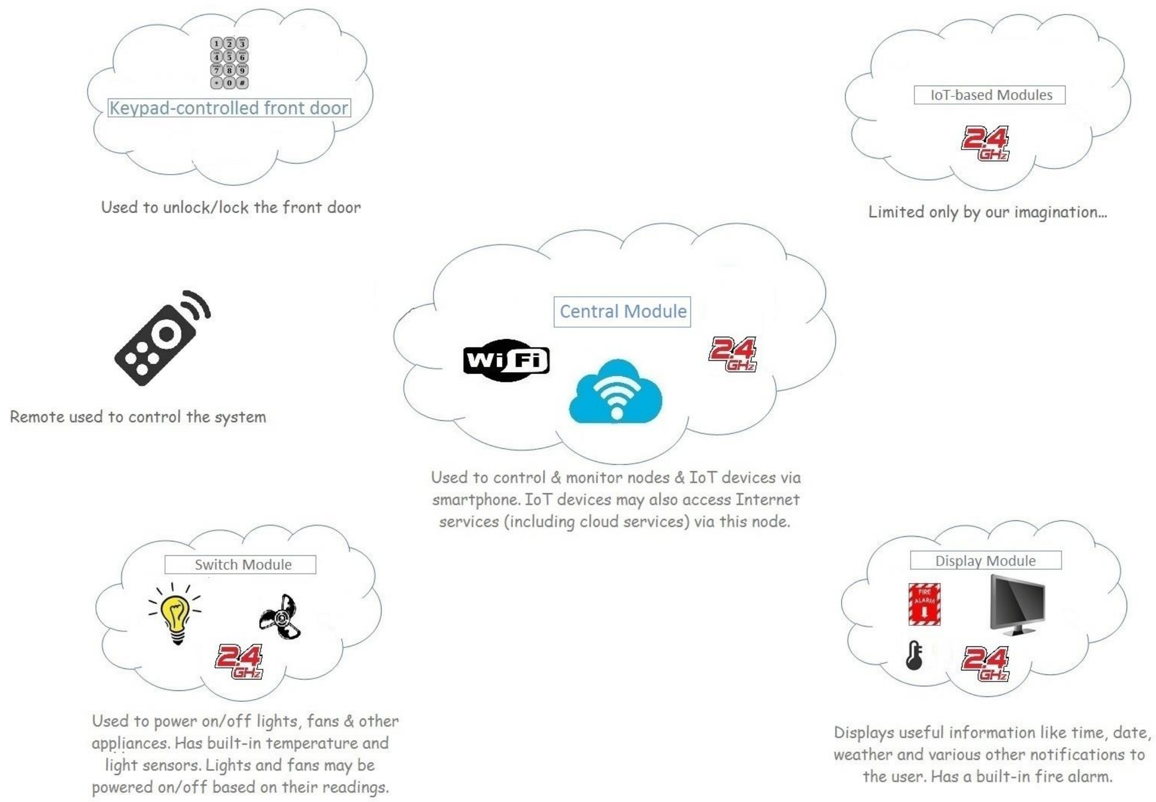


Fig. 3.1 Block diagram of the system

3.1 DESIGN OF THE MICROCONTROLLER (MCU) BASED MODULES

All MCU-based modules have been implemented using the PIC18F4550. The PIC18 family of MCUs were selected due to the following reasons:

- Extremely reliable architecture.
- User-friendly IDE (Integrated Development Environment) and professional-grade C-compiler.
- Excellent documentation and programming tools.
- Easily available at lower price (compared to rival manufacturers).
- The rich set of features and peripherals on-board the PIC18F4550 allows plenty of room for further functionality improvement and enhancement.

Specifications of the PIC18F4550:

- 8-bit MCU.
- 32kB flash memory.
- 2kB RAM.
- 256 bytes EEPROM.
- Available in a 40 pin DIP (Dual-inline package).
- 12MIPS maximum throughput.

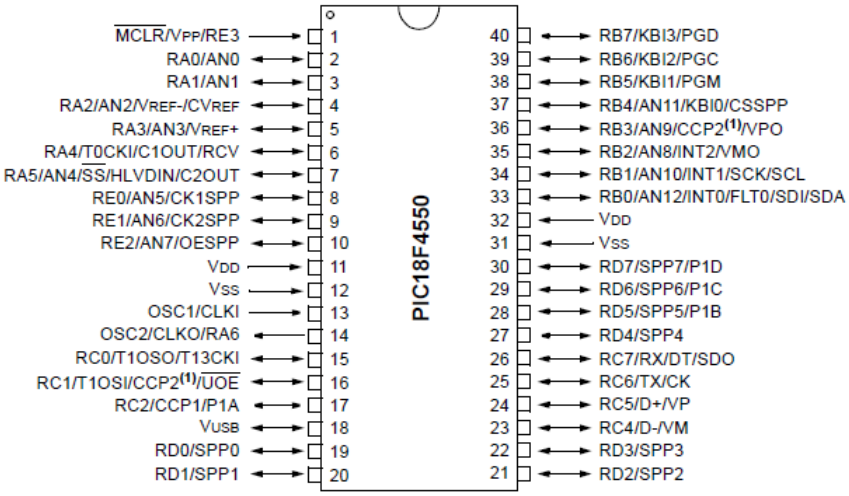


Fig. 3.2 PIC18F4550 pin diagram

3.1.1. Wireless Module Interfacing

While Zigbee is a good choice for this application, it was too expensive. It was decided to use the low cost nRF24L01+ 2.4GHz RF transceivers by Nordic Semiconductors. nRF24L01+ has been specifically designed for power efficient IoT-based applications which require efficient and reliable communication and can be easily interfaced to an MCU.

Basic specifications of the nRF24L01+:

- Communication at 2.4GHz with air data rates of up to 2Mbps.
- SPI interface.
- Maximum data payload of 32 bytes.
- 3.3V logic (5V-tolerant I/O pins).

▪ Interrupt on packet reception and transmission.

▪ Guaranteed 30m line-of-sight range at 2Mbps air data rate.

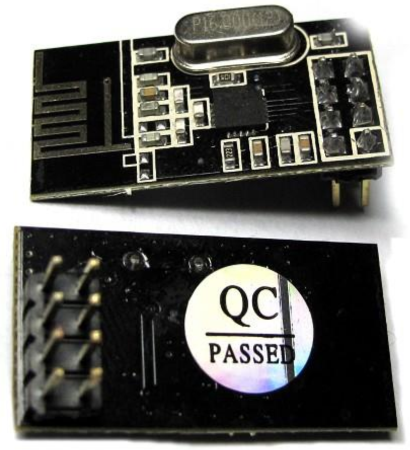


Fig. 3.3 nRF24L01+ module

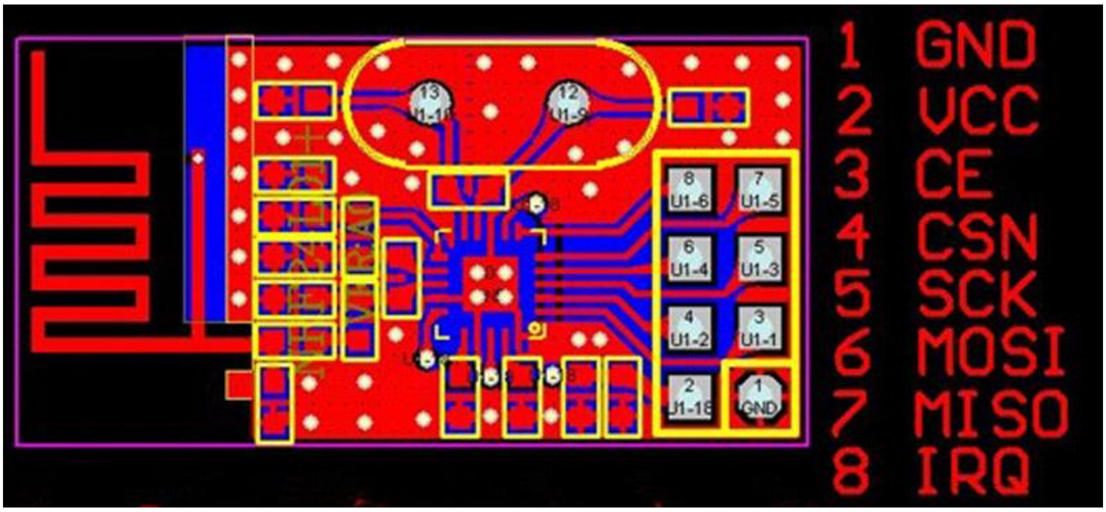


Fig. 3.4 Pin diagram of an nRF24L01+ module

With plenty of documentation and tutorials available for it, it was easy to get an Arduino and a PIC MCU communicating with each other wirelessly in the tests. In the project, the data rate has been set to 250kbps (slowest) and the payload length to 8 bytes. The SPI bus speed is limited to 312.5 kbps. The module can only transmit or receive data at any given instant (half-duplex communication).

The device has 5 different operation modes:

Mode	PWR_UP register	PRIM_RX register	CE input pin	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFOs. Will empty all levels in TX FIFOs[a].
TX mode	1	0	Minimum 10µs high pulse	Data in TX FIFOs.Will empty one level in TX FIFOs[b].
Standby-II	1	0	1	TX FIFO empty.
Standby-I	1	-	0	No ongoing packet transmission.
Power Down	0	-	-	-

Table 3.1 Operating modes of an nRF24L01+ module

Given below is the finite state machine diagram radio control of the nRF24L01+:

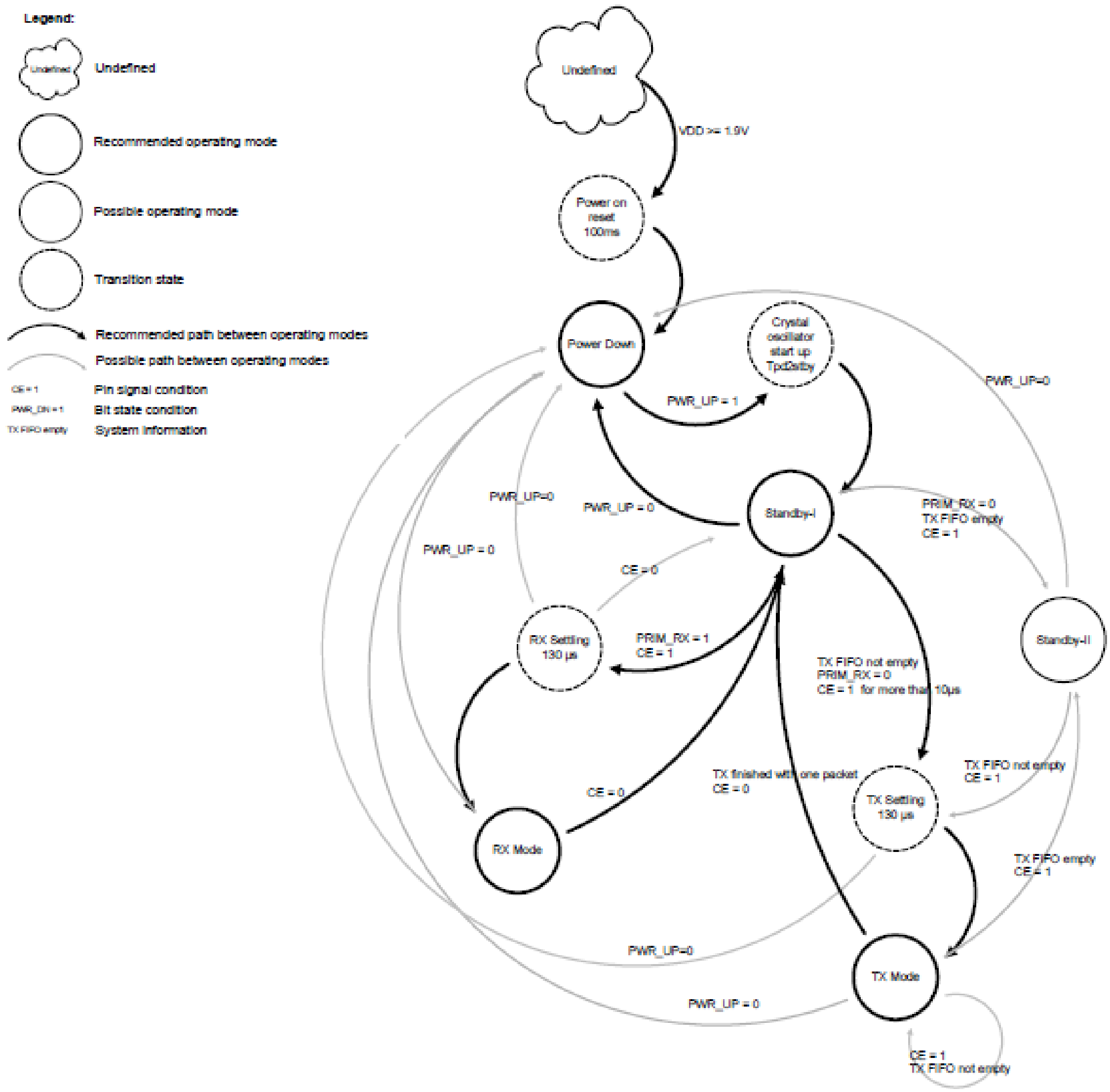


Fig. 3.5 Radio control state diagram of nRF24L01+ module

3.1.2 Relay Module

Relay module is implemented using a PIC18F4550. Testing involved sending control codes via the remote controller and checking for the correct response from the module. All communication took place wirelessly via the nRF24L01+.

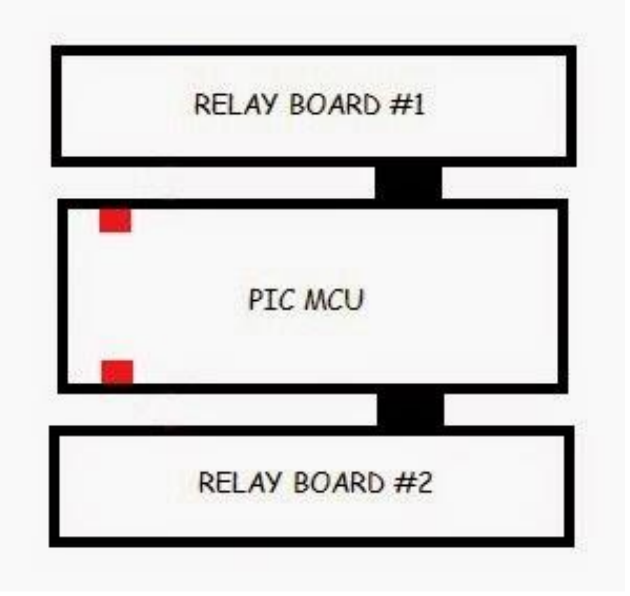


Fig. 3.6 Block diagram of the relay module

- There are 8 relays on each board (16 relays in total). The end-user may choose to buy 1 or both boards for use in the system. Off-the-shelf relay boards will be used in the design.
- The black connections between the PIC MCU and the relay boards represent control lines for energizing/de-energizing the relays.
- The red dot on the PIC MCU board represents enable lines for each of the board. This is a safety feature.
- Whenever a relay board is connected, the enable line for it is activated. This is detected by the PIC MCU. The ports controlling that relay board can only be used when the enable line is activated.

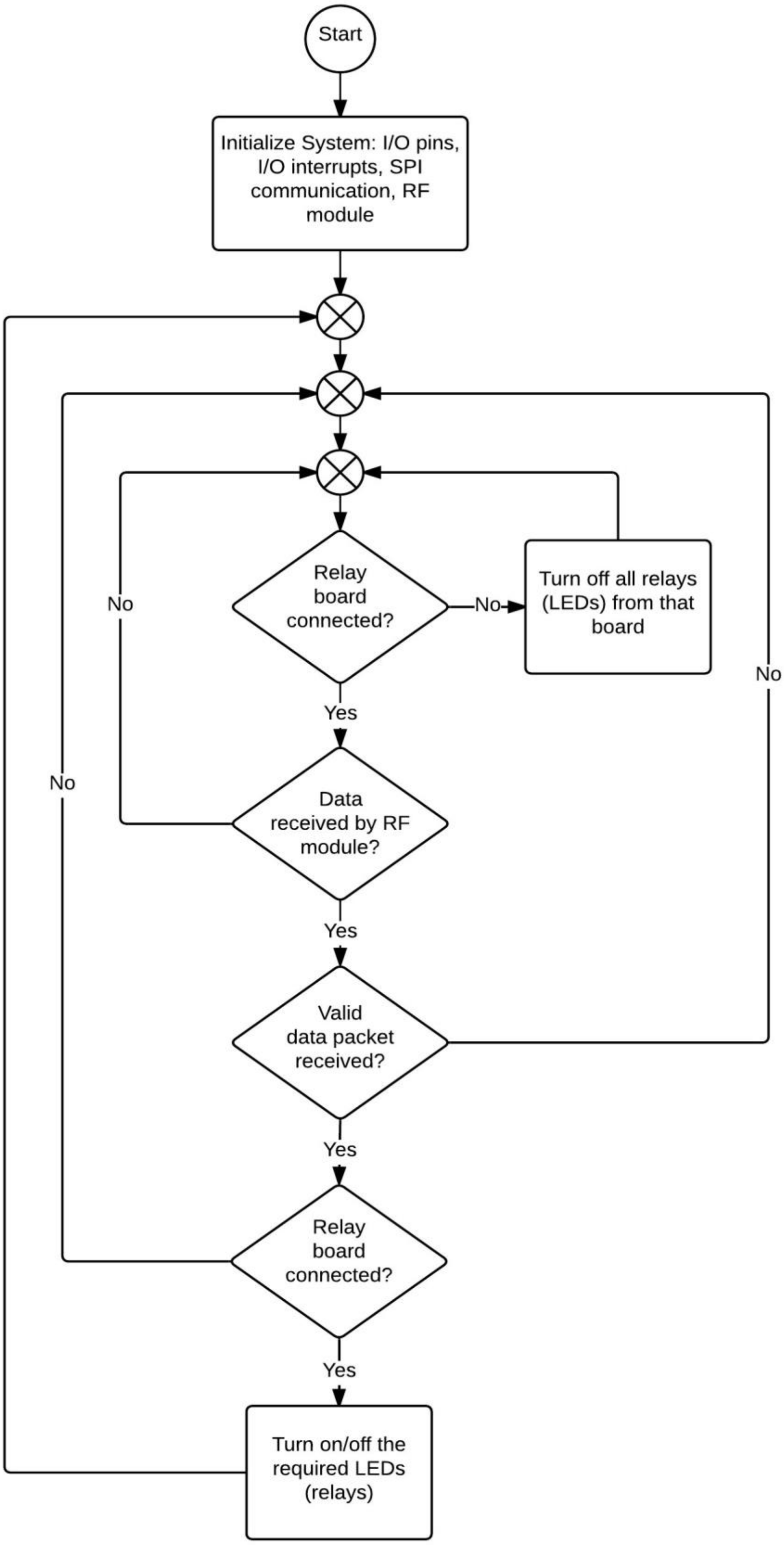


Fig. 3.7 Flowchart of the relay module program

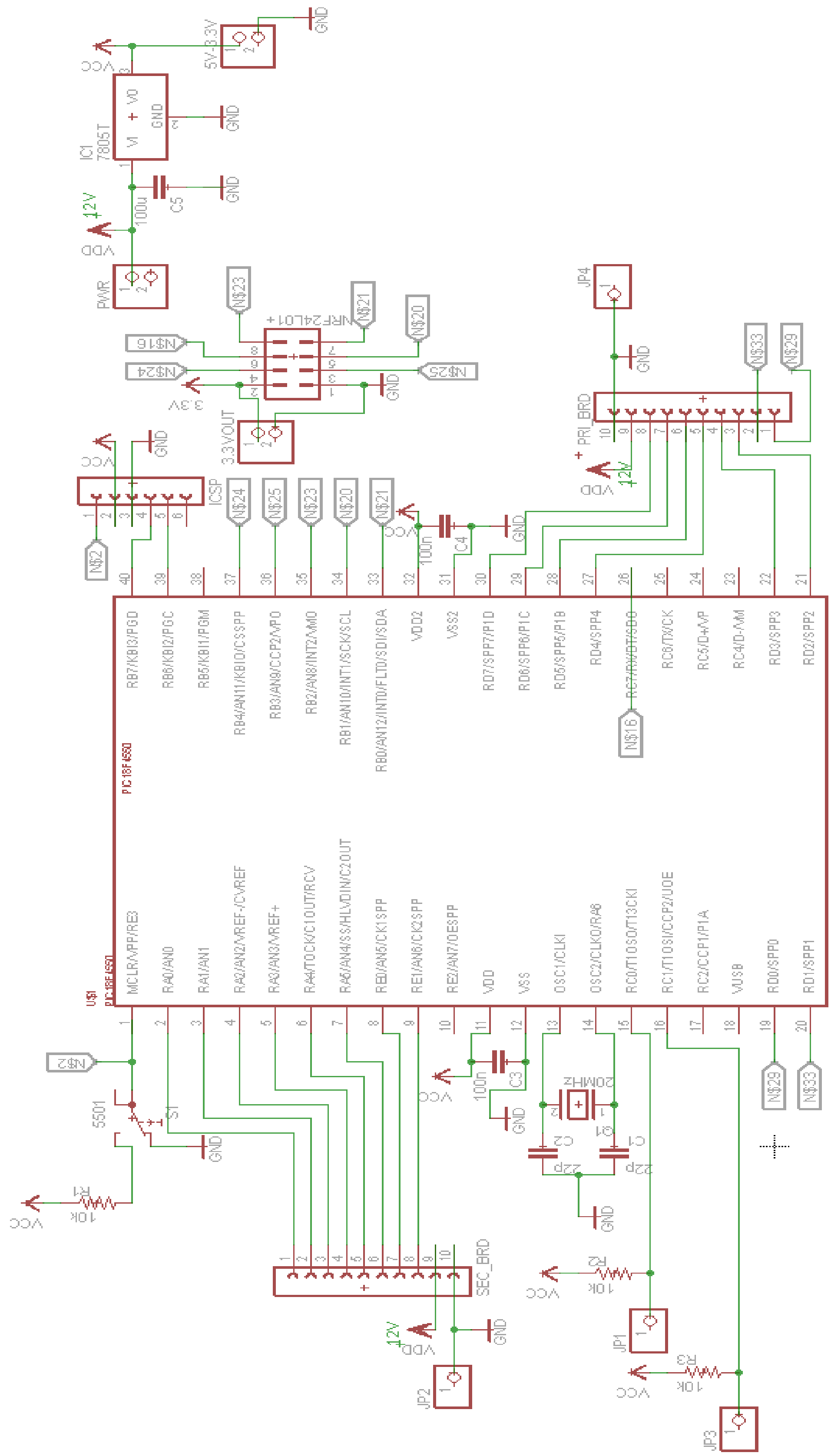


Fig. 3.8 Circuit schematic of the relay module

3.1.3 Display Module

The display module has been implemented using the PIC18F4550. There are 2 types of messages which this module will be receiving wirelessly via the nRF24L01+:

- Weather data
- Time & date data

The module toggles between displaying time-date and weather every 10 seconds. This information is displayed on a standard Hitachi HD44780-controlled 16x2 alphanumeric LCD. This LCD is widely used in embedded system projects and hence, it was easy to interface it with the PIC MCU.

An LM35 temperature sensor is also used for sensing the room temperature. An indicator LED (alarm) will light up to indicate the possibility of fire if the temperature exceeds 55°C. Just like the LCD, LM35 is a widely used temperature sensor in embedded system projects and it was easy to interface it with the PIC MCU. It is factory-calibrated directly in °C (10mV/°C linear sensitivity) and can be used to sense temperature ranging from -55°C to 150°C.

Hence there are 2 parts to the program (both running simultaneously):

- Display management: Toggles between displaying time-date and weather.
- Wireless communication: Receives data and sorts it systematically for displaying.

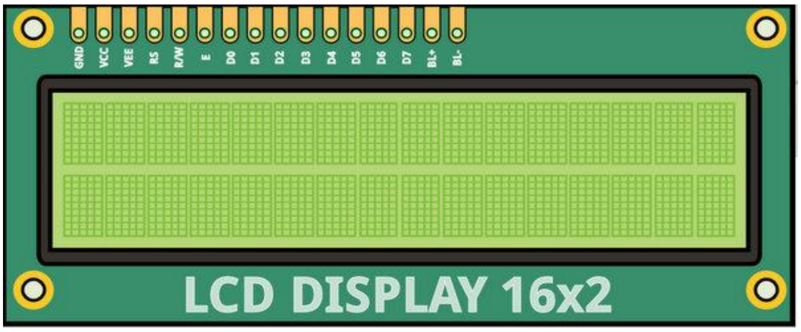


Fig. 3.9 Hitachi HD44780 controlled 16x2 alphanumeric LCD module pin diagram

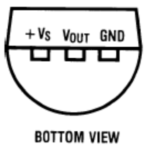


Fig. 3.10 LM35 temperature sensor pin diagram

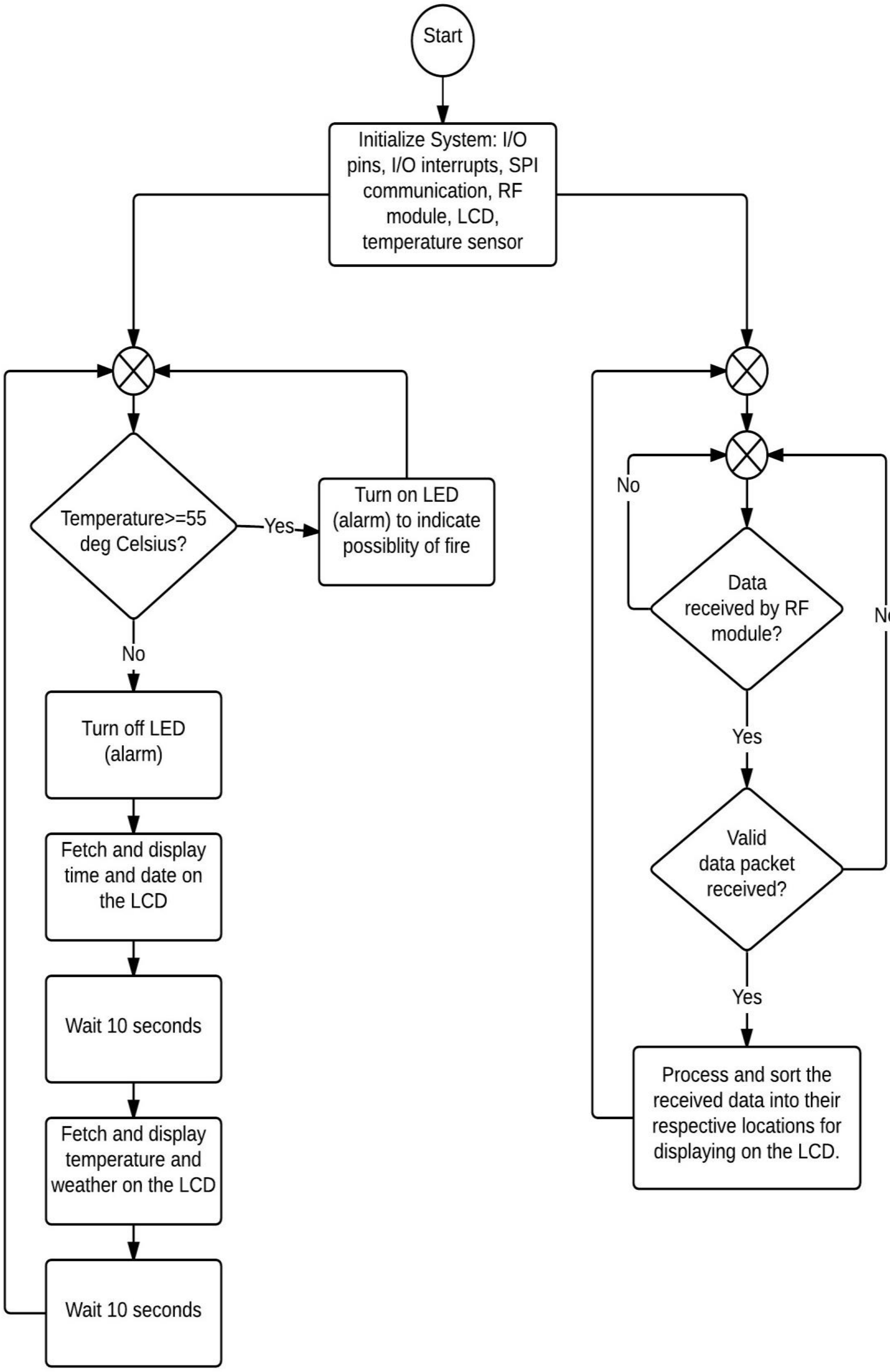


Fig. 3.11 Flowchart of the display module program

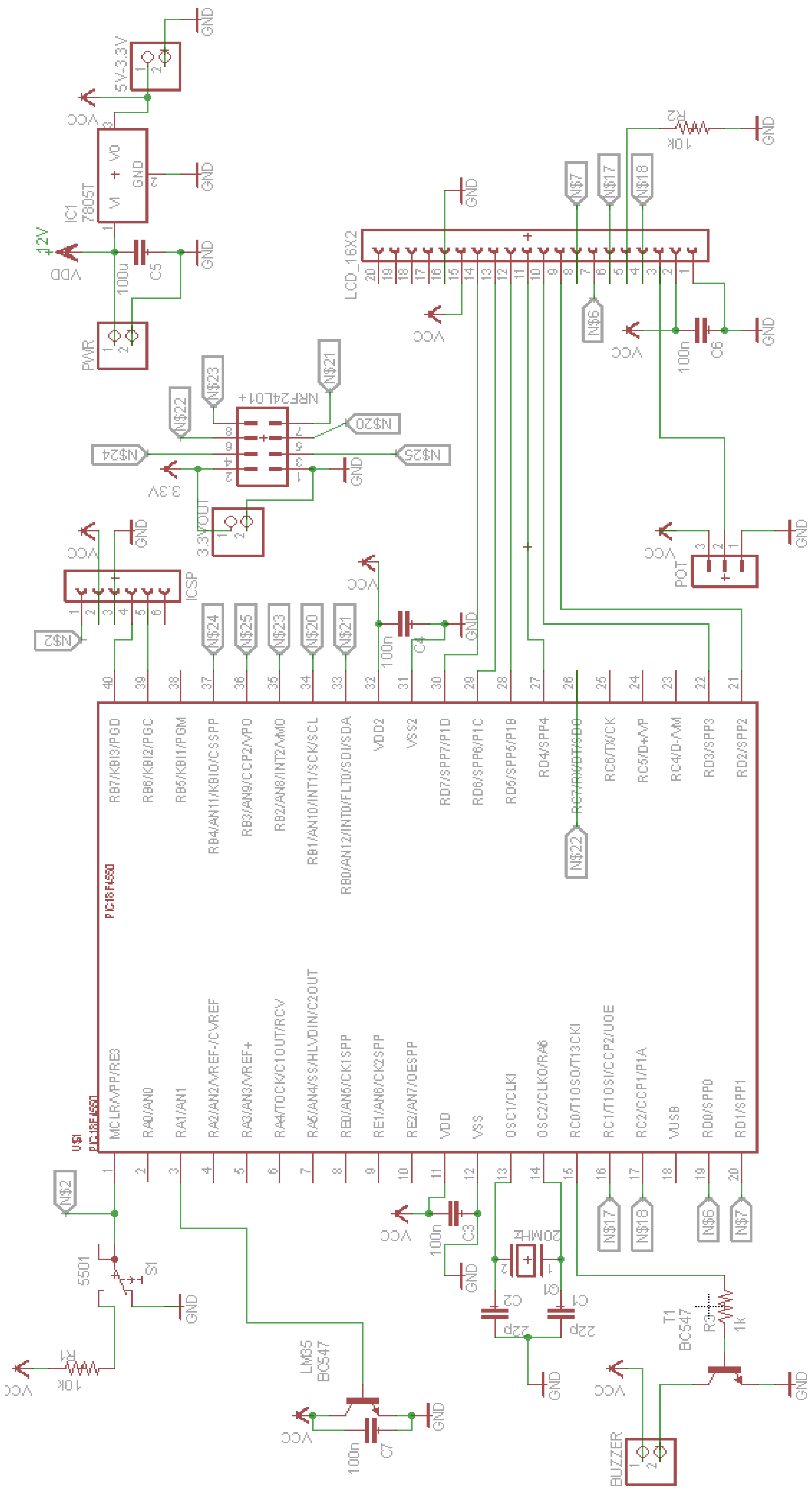


Fig. 3.12 Circuit schematic of the display module

3.1.4 Keypad-Based Door Control Module

The door control module has been implemented using a PIC18F4550. This will be the only module in the project which does not have a wireless module (nRF24L01+) attached to it. A standard Hitachi HD44780-controlled 16x2 alphanumeric LCD is used to display messages to and interact with the user. This LCD is widely used in embedded system projects and hence, it was easy to interface it with the PIC MCU.

A standard 4x4 matrix keypad is used to enter the password.

Features of this module:

- Standard Hitachi HD44780 controlled 16x2 alphanumeric LCD displays useful messages to the user.
- 4x4 keypad used to enter passwords.
- A yellow LED used to indicate key press.
- A green LED used to indicate that the door lock actuator is energized.
- Pushbutton used to start/stop password change procedure.
- Password is stored in the internal EEPROM of the PIC MCU.

When a correct password is entered, the actuator is energized for 5 seconds. The user must open the door in this time or he/she would have to enter the password again. Password length has been set to 4 digits (can be easily set to any value up to 256 digits).

For changing the password, the password-change button is pressed. Standard 3-step password-changing procedure is used to change the password:

- Entry of old password
- Entry of new password
- Confirmation of new password

The user can choose to quit the password-changing procedure by pressing the password-change button at any point of time in between if he decides against changing the password.

Hence there are 2 parts to the program, both running simultaneously on the MCU:

- **Password authentication:** Used for door access control.
- **Password change control:** Used to change the password used for door access control.

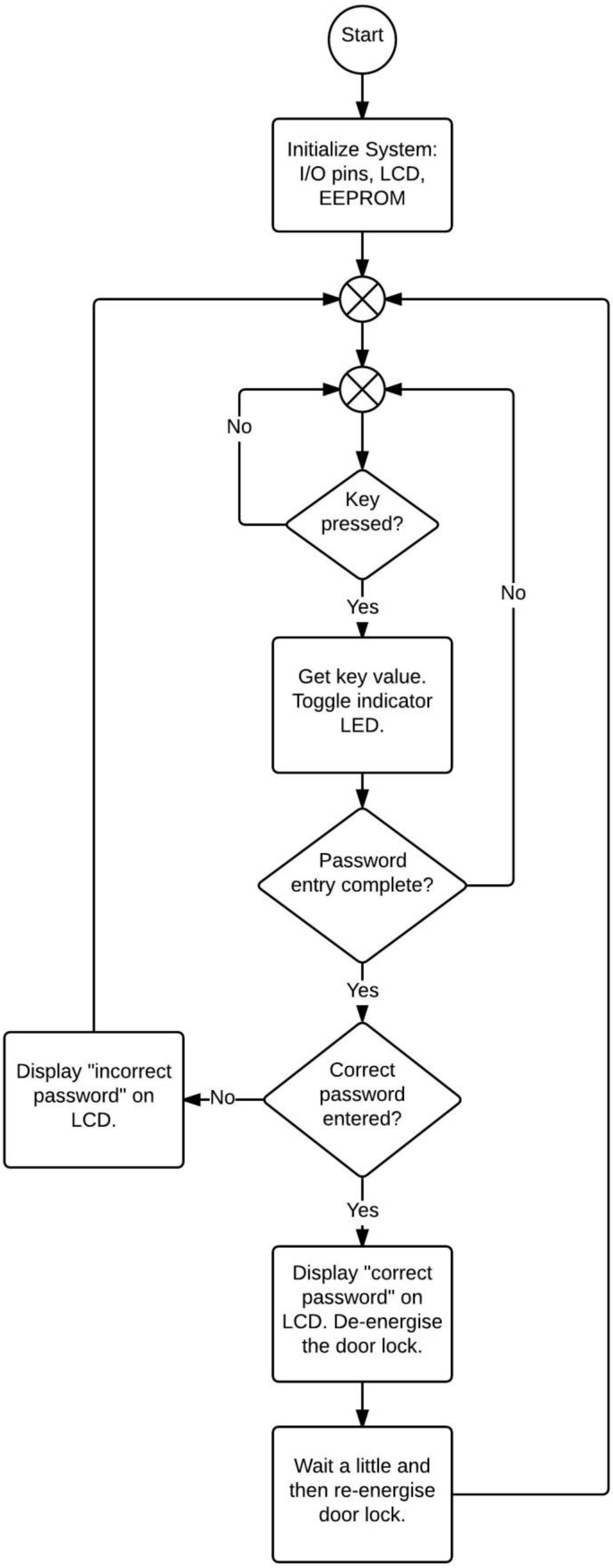


Fig. 3.13 Flowchart of the password authentication sub-program

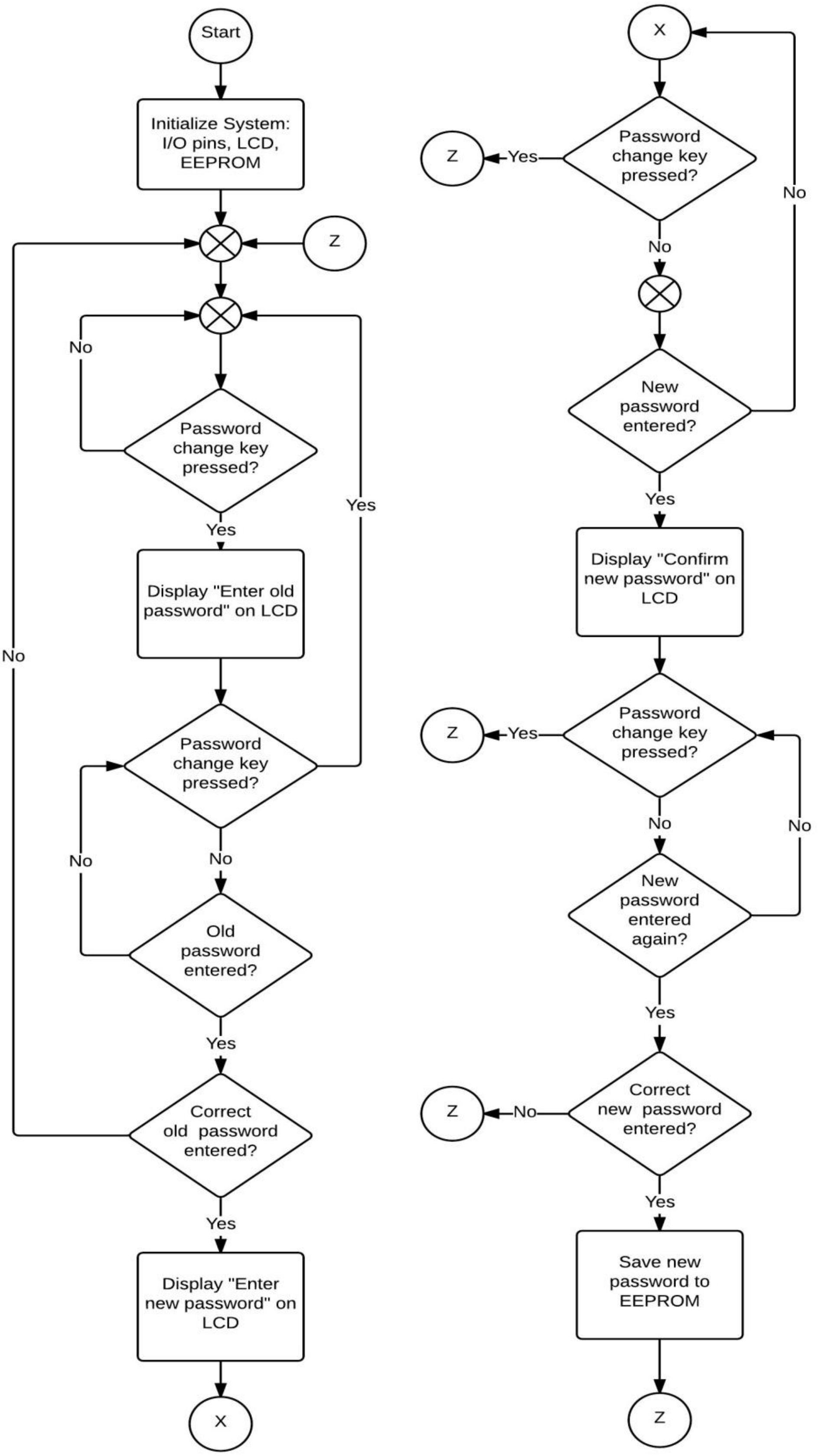


Fig. 3.14 Flowchart of the password change control sub-program

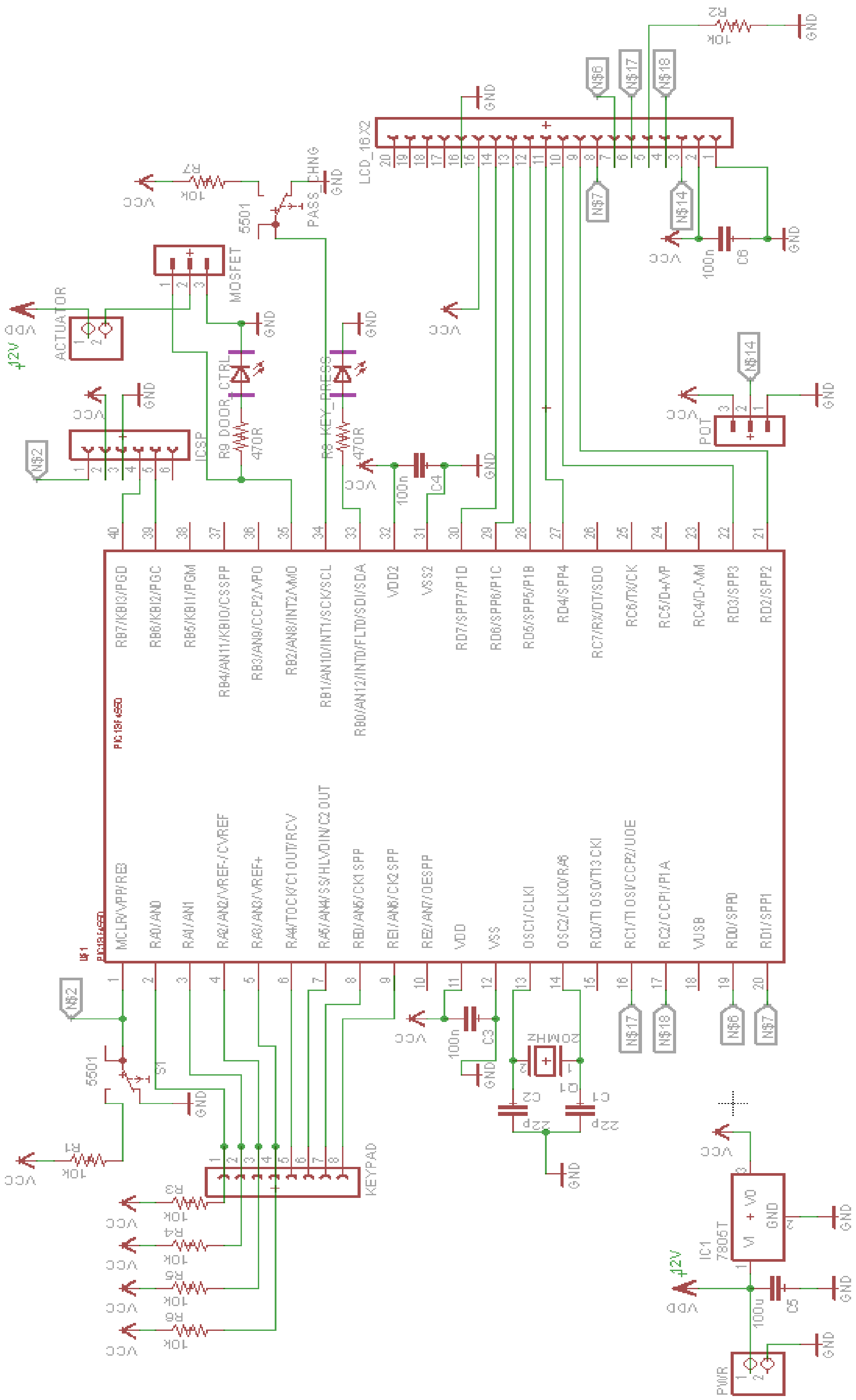


Fig. 3.15 Circuit schematic of the door control module

3.2 DESIGN OF THE CENTRAL MODULE

The central module is being implemented using a Raspberry Pi B+ single-board computer (SBC) and an Arduino Uno.

Key specifications of the Raspberry Pi B+:

- Broadcom BCM2835 SoC
- 700MHz ARM11 CPU
- 512MB RAM
- 4 USB 2.0 ports, 1 RJ45 Ethernet port, 1 HDMI port, 1 3.5mm audio jack
- 20x2 pin header

Fig. 3.16 Raspberry Pi Model B+

The Raspberry Pi is a widely used versatile and powerful Linux-based SBC with extensive documentation and community support. Since Raspbian (the default Linux “distro” for the Raspberry Pi) is Debian Wheezy-based, most of the standard Linux libraries can be used out-of-the-box without any modifications. Use of Python for programming is an additional advantage due to its ease of use.

Key specifications of the Arduino Uno:

- ATmega328 8-bit MCU.
- 32kB flash memory.
- 2kB RAM.
- 1kB EEPROM
- Available in a 28-pin DIP.
- 16MIPS maximum throughput.

Fig. 3.17 Arduino Uno

The Arduino Uno is another widely used versatile and powerful open-source MCU development board with extensive community support and documentation.

Weather data is fetched through standard developer APIs from Weather Underground. The data is received in JavaScript format, which is then processed by the Raspberry Pi using standard Python libraries into a structured dictionary for easy access. Date and time are fetched and processed using standard Python libraries as well.

These are sent to the Arduino Uno via UART (Universal Asynchronous Receiver Transmitter). The Arduino then transmits this information wirelessly via nRF24L01+.

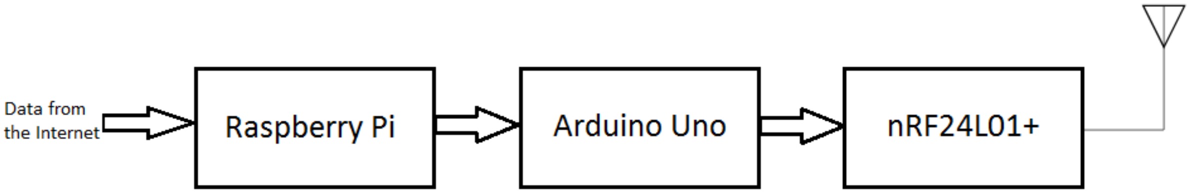


Fig. 3.18 Flow of data on the central module

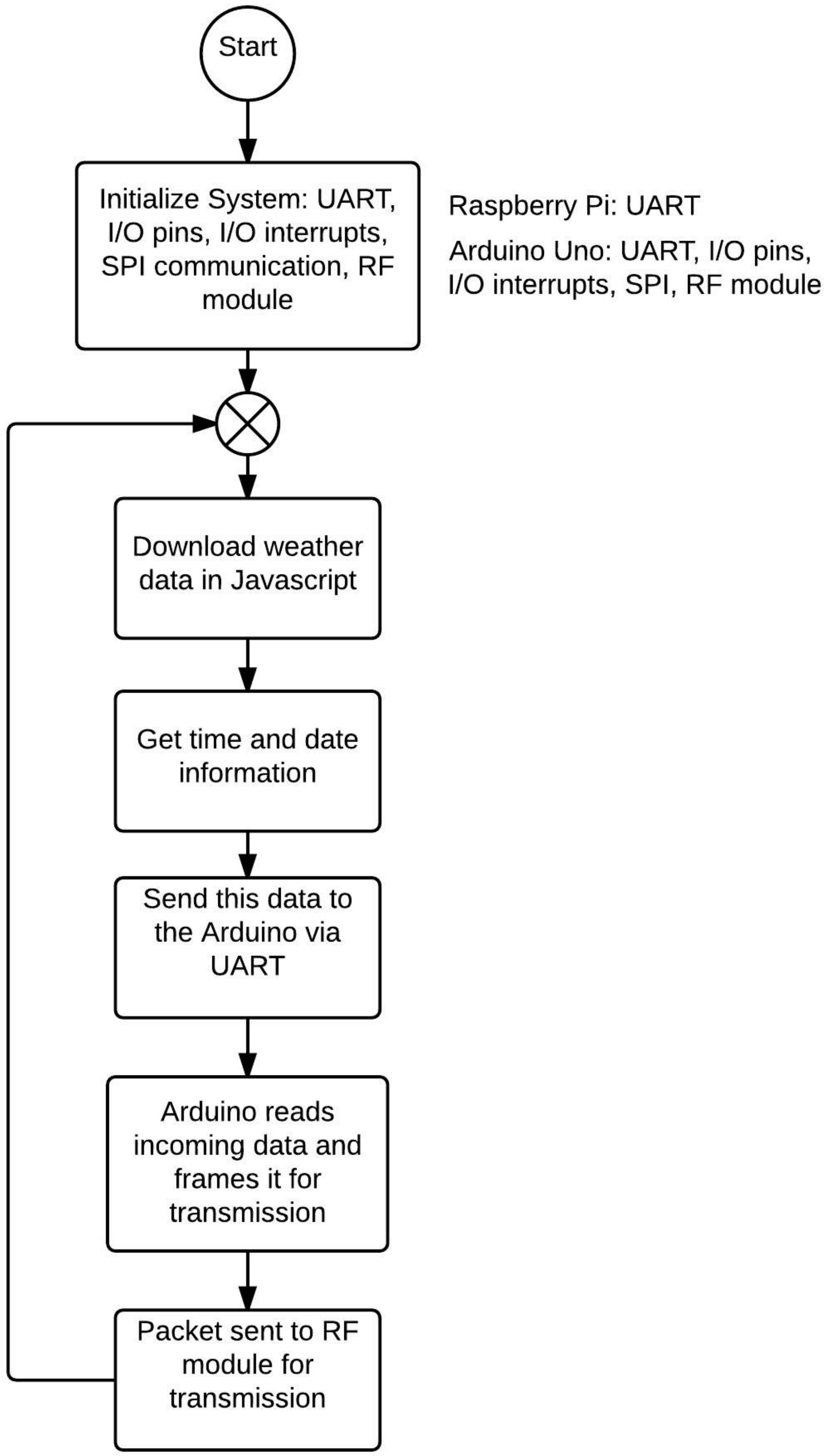


Fig. 3.19 Flowchart of the central module program

3.3 DESIGN OF THE REMOTE CONTROLLER

The central module is being implemented using a Raspberry Pi B+ single-board computer (SBC) and an Arduino Mega 2560. An Arduino and a Raspberry Pi was chosen again here mainly due to the large community support and extensive documentation for each of the platforms and their flexibility.

Key specifications of the Arduino Mega 2560:

- ATmega 2560 8-bit MCU.
- 256kB flash memory.
- 8kB RAM.
- 4kB EEPROM
- 54 GPIO (general purpose I/O) pins
- 16MIPS maximum throughput.

Fig. 3.20 Arduino Mega 2560

A standard Python GUI developer framework known as TKinter to develop the GUI in Python. The GUI has been designed to be simple as well as maintain functionality and reliability. The GUI has been designed for controlling 1 relay boards for now. It can easily be expanded for controlling more boards if necessary. The GUI is intended for touch-based controllers but can be used through a simple mouse-keyboard interface too.

This Python-based GUI accepts inputs from the user and controls a number of I/O pins accordingly. The Arduino Mega 2560 reads the state of these I/O pins and accordingly sends control signals to the relay module.

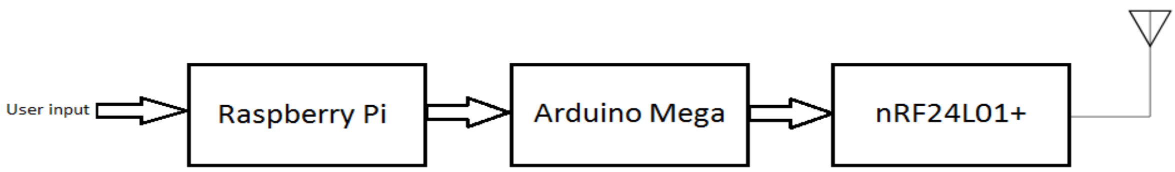


Fig. 3.21 Flow of data on the remote controller

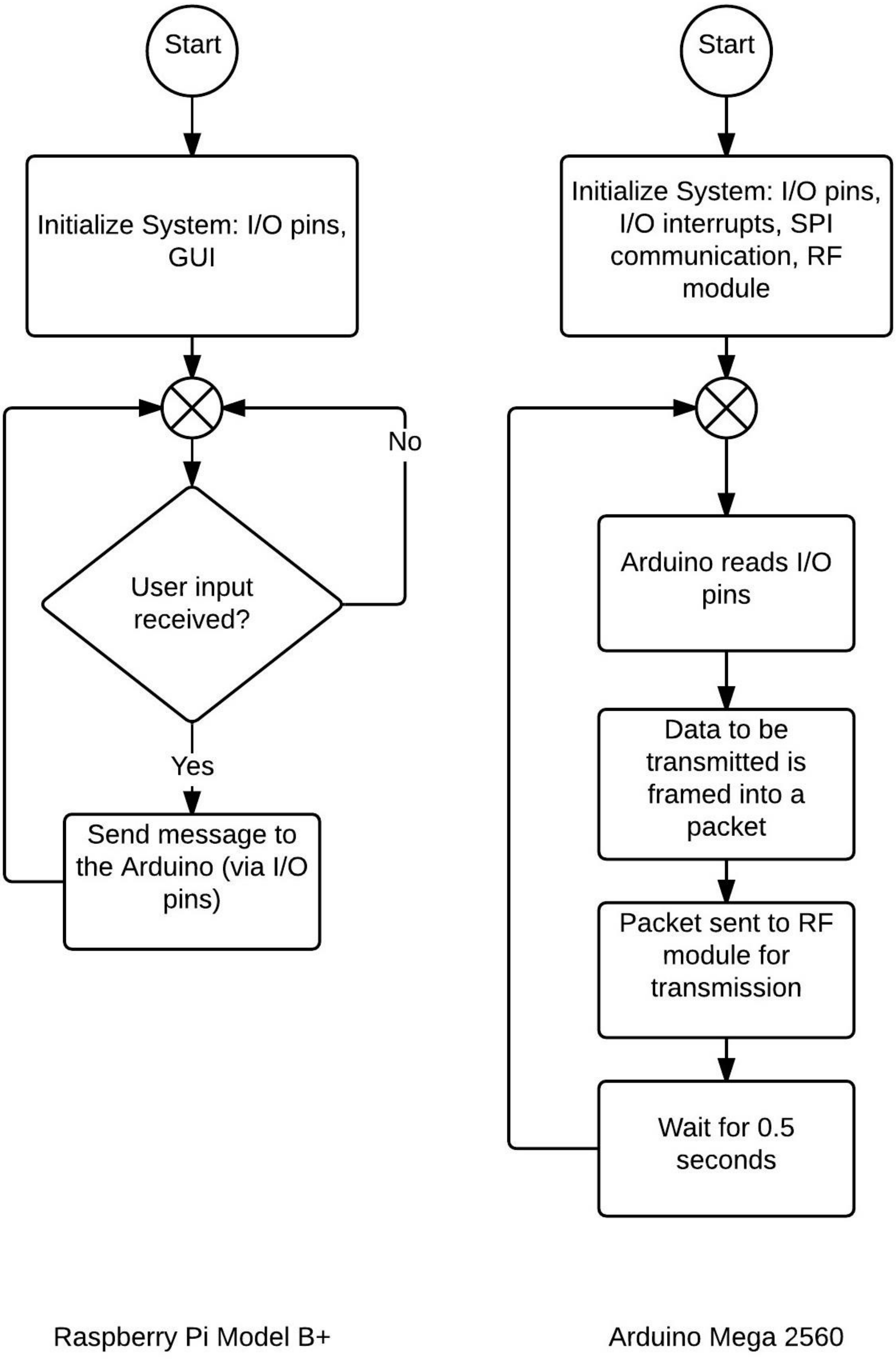


Fig. 3.22 Flowchart of the remote controller program

3.4 POWER SUPPLY DESIGN

The system requires 2 separate power rails for operation:

- 5V
- 3.3V (only for the nRF24L01+)

The 5V power rail for the MCU-based modules was designed using a 7805 positive voltage regulator. It is a widely used voltage regulator with maximum output current of up to 500mA (without heat sink) and up to 1A (with heat sink), more than enough to ensure smooth operation of the system.

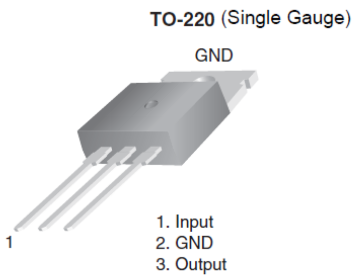


Fig. 3.23 7805 pin diagram

The 3.3V power rail for the MCU-based modules was designed using an AMS1117 LDO (low drop-out) voltage regulator. 5V output from the 7805 is fed into the AMS1117 to create the 3.3V rail for the nRF24L01+. It can provide output current of up to 1A. Since the IC is available only in SMD (surface-mount device) packages, a breakout PCB was used for implementation as shown below.

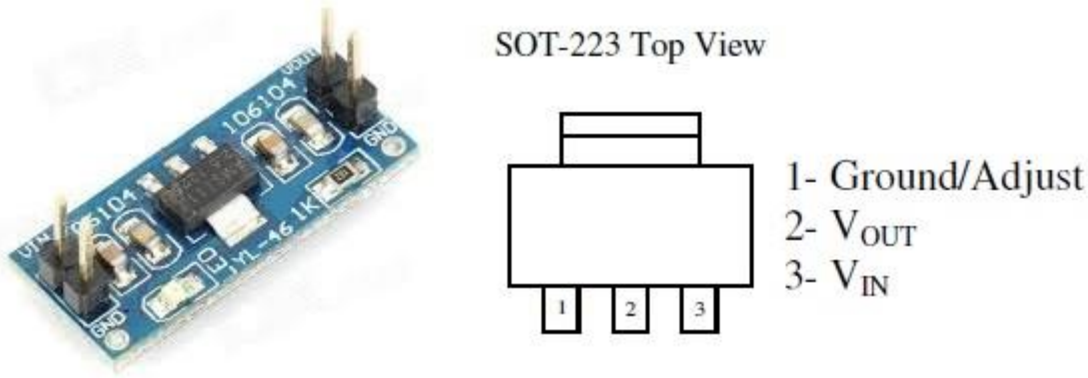


Fig. 3.24 AMS1117 LDO voltage regulator

The remote controller and the central modules already have 5V and 3.3V power rails built into them.

CHAPTER 4
RESULT ANALYSIS

With all the modules independently designed and tested successfully, they were finally integrated into 1 complete system. The following glitches/bugs/shortcomings where noticed during the final integration and testing:

- A glitch was noticed in the relay module which caused the relays to automatically switch on/off erratically. This was caused due to the program continuously scanning and updates the I/O pin states. This was corrected to only scan/update the I/O pins when it receives a new data packet through the nRF24L01+.

- An additional shortcoming discovered in the prototype was the high consumption of power by the modules (IoT-based applications should be power efficient) as they were fully active even when not doing any significant work. This could not be rectified at this stage without changing the program flow significantly and conducting extensive tests on the performance of the new code. This can, however be corrected in future iterations of the prototype and the features required to achieve this are already present in the processing devices (as discussed in the next chapter).

- A glitch was observed in the Arduino Uno development board which is connected to the display module. The latest revision of the Arduino Uno boot-loader has a bug which sometimes causes the MCU to “hang up” during operation. This will however be rectified in forthcoming versions of the boot-loader. A temporary fix to this problem would be to use another Arduino Mega 2560 instead of an Arduino Uno (the boot-loader for the Arduino Mega 2560 is completely stable) or flashing the ATmega328 on the Arduino Uno with a previous version of the boot-loader.

Rest assured, the system performed as expected during the tests.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE OF WORK

The objective was to design and implement a basic and functional home automation system which allows easy integration of IoT-enabled devices.

A modular approach was employed to design the system (described earlier in detail):

- Design of the microcontroller based modules.
- Design of the remote controller (used by the user to control the system).
- Design of the central module.
- Final integration and testing.

While a basic and functional home automation system was successfully designed, implemented and tested, there is plenty of space for future improvements in the design:

- **Increasing power efficiency:** Improvement in power consumption by put the modules in sleep mode and using wake-up-on-interrupt feature (available in most MCUs now). This would result in a drastic decline in power consumption. After all, power efficiency is envisioned to be a defining characteristic of most IoT-enabled devices.
- **Use of smartphone based remote controllers:** Carrying an extra device can tend to be cumbersome and tedious for the end-user. A smartphone-based solution on the other hand is more efficient and handy. Suitable apps can be designed for Android and iOS based devices (since these are the 2 most widely used mobile OS platforms). This would however require that all communication to and from the smartphone occur through the central module as that is the only module connected to the Internet directly (Bluetooth has limited range). Used of a smartphone will also allow easy control of any IoT-enabled devices connected to the system. A whole ecosystem of apps for control of the system and IoT-enabled devices can be created.
- **Use of a more aesthetically appealing GUI on the display module:** This would however increase the cost of the system as an SBC would be required to support such a GUI.
- **Use of camera at the front door: To** take a snapshot of the person who is entering the house. This can be used for security by alerting the user to possible break-in. Various other security measures like motion sensors can also be used at other doors and windows.

REFERENCES

1. **O. Monnier**, A smarter grid with the Internet of Things, SLYB214, Texas Instruments, May 2014.

2. **Harper, Richard,** Inside the smart home, Springer, ISBN: 1852336889

3. **Mann, William,** Smart technology for aging, disability and independence: the state of the science, John Wiley & Sons, ISBN: 0471696943

ANNEXURE

NORDIC SEMICONDUCTOR

nRF24L01+

Single Chip 2.4GHz Transceiver

Product Specification v1.0

Key Features

- Worldwide 2.4GHz ISM band operation
- 250kbps, 1Mbps and 2Mbps on air data rates
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 13.5mA RX at 2Mbps air data rate
- 900nA in power down
- 26µA in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with nRF24L01
- On-air compatible in 250kbps and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
- Low cost BOM
- ±60ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

Applications

- Wireless PC Peripherals
- Mouse, keyboards and remotes
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Game controllers
- Sports watches and sensors
- RF remote controls for consumer electronics
- Home and commercial automation
- Ultra low power sensor networks
- Active RFID
- Asset tracking systems
- Toys

September 2008

9 Register Map

You can configure and control the radio by accessing the register map through the SPI.

9.1 Register map table

All undefined bits in the table below are redundant. They are read out as '0'.

Note: Addresses 18 to 1B are reserved for test purposes, altering them makes the chip malfunction.

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' – 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0:POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control 1: PRX, 0: PTX
01	EN_AA Enhanced ShockBurst™				Enable 'Auto Acknowledgment' Function Disable this functionality to be compatible with nRF2401, see page 75
	Reserved	7:6	00	R/W	Only '00' allowed
	ENAA_P5	5	1	R/W	Enable auto acknowledgement data pipe 5
	ENAA_P4	4	1	R/W	Enable auto acknowledgement data pipe 4
	ENAA_P3	3	1	R/W	Enable auto acknowledgement data pipe 3
	ENAA_P2	2	1	R/W	Enable auto acknowledgement data pipe 2
	ENAA_P1	1	1	R/W	Enable auto acknowledgement data pipe 1
	ENAA_P0	0	1	R/W	Enable auto acknowledgement data pipe 0
02	EN_RXADDR				Enabled RX Addresses
	Reserved	7:6	00	R/W	Only '00' allowed
	ERX_P5	5	0	R/W	Enable data pipe 5.
	ERX_P4	4	0	R/W	Enable data pipe 4.
	ERX_P3	3	0	R/W	Enable data pipe 3.
	ERX_P2	2	0	R/W	Enable data pipe 2.

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
	ERX_P1	1	1	R/W	Enable data pipe 1.
	ERX_P0	0	1	R/W	Enable data pipe 0.
03	SETUP_AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' – 5 bytes LSByte is used if address width is below 5 bytes
04	SETUP_RETR				Setup of Automatic Retransmission
	ARD[a]	7:4	0000	R/W	Auto Retransmit Delay '0000' – Wait 250µS '0001' – Wait 500µS '0010' – Wait 750µS …….. '1111' – Wait 4000µS (Delay defined from end of transmission to start of next transmission)[b]
	ARC	3:0	0011	R/W	Auto Retransmit Count '0000' –Re-Transmit disabled '0001' – Up to 1 Re-Transmit on fail of AA …… '1111' – Up to 15 Re-Transmit on fail of AA
05	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel nRF24L01+ operates on
06	RF_SETUP				RF Setup Register
	CONT_WAVE	7	0	R/W	Enables continuous carrier transmit when high.
	Reserved	6	0	R/W	Only '0' allowed
	RF_DR_LOW	5	0	R/W	Set RF Data Rate to 250kbps. See RF_DR_HIGH for encoding.
	PLL_LOCK	4	0	R/W	Force PLL lock signal. Only used in test
	RF_DR_HIGH	3	1	R/W	Select between the high speed data rates. This bit is don't care if RF_DR_LOW is set. Encoding: [RF_DR_LOW, RF_DR_HIGH]: '00' – 1Mbps '01' – 2Mbps '10' – 250kbps '11' – Reserved

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
	RF_PWR	2:1	11	R/W	Set RF output power in TX mode '00' – -18dBm '01' – -12dBm '10' – -6dBm '11' – 0dBm
	Obsolete	0			Don't care
07	STATUS				Status Register (In parallel to the SPI command word applied on the MOSI pin, the STATUS register is shifted serially out on the MISO pin)
	Reserved	7	0	R/W	Only '0' allowed
	RX_DR	6	0	R/W	Data Ready RX FIFO interrupt. Asserted when new data arrives RX FIFO[c]. Write 1 to clear bit.
	TX_DS	5	0	R/W	Data Sent TX FIFO interrupt. Asserted when packet transmitted on TX. If AUTO_ACK is activated, this bit is set high only when ACK is received. Write 1 to clear bit.
	MAX_RT	4	0	R/W	Maximum number of TX retransmits interrupt Write 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.
	RX_P_NO	3:1	111	R	Data pipe number for the payload available for reading from RX_FIFO 000-101: Data Pipe Number 110: Not Used 111: RX FIFO Empty
	TX_FULL	0	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
08	OBSERVE_TX				Transmit observe register
	PLOS_CNT	7:4	0	R	Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to RF_CH. See page 75.
	ARC_CNT	3:0	0	R	Count retransmitted packets. The counter is reset when transmission of a new packet starts. See page 75.
09	RPD				
	Reserved	7:1	000000	R	
	RPD	0	0	R	Received Power Detector. This register is called CD (Carrier Detect) in the nRF24L01. The name is different in nRF24L01+ due to the different input power level threshold for this bit. See section 6.4 on page 25.
0A	RX_ADDR_P0	39:0	0xE7E7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
0B	RX_ADDR_P1	39:0	0xC2C2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0C	RX_ADDR_P2	7:0	0xC3	R/W	Receive address data pipe 2. Only LSB. MSBytes are equal to RX_ADDR_P1[39:8]
0D	RX_ADDR_P3	7:0	0xC4	R/W	Receive address data pipe 3. Only LSB. MSBytes are equal to RX_ADDR_P1[39:8]
0E	RX_ADDR_P4	7:0	0xC5	R/W	Receive address data pipe 4. Only LSB. MSBytes are equal to RX_ADDR_P1[39:8]
0F	RX_ADDR_P5	7:0	0xC6	R/W	Receive address data pipe 5. Only LSB. MSBytes are equal to RX_ADDR_P1[39:8]
10	TX_ADDR	39:0	0xE7E7E7E7E7	R/W	Transmit address. Used for a PTX device only. (LSByte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device with Enhanced ShockBurst™ enabled. See page 75.
11	RX_PW_P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P0	5:0	0	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte … 32 = 32 bytes
12	RX_PW_P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P1	5:0	0	R/W	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte … 32 = 32 bytes
13	RX_PW_P2				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P2	5:0	0	R/W	Number of bytes in RX payload in data pipe 2 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte … 32 = 32 bytes
14	RX_PW_P3				
	Reserved	7:6	00	R/W	Only '00' allowed

National Semiconductor

November 2000

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

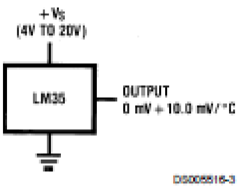
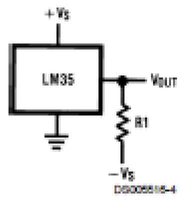


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$

$V_{OUT} = +1{,}500$ mV at +150°C
$= +250$ mV at +25°C
$= -550$ mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

 DS005516 www.national.com

PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1 Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Idle mode Currents Down to 5.8 μA Typical
- Sleep mode Currents Down to 0.1 μA Typical
- Timer1 Oscillator: 1.1 μA Typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μA Typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High-Precision PLL for USB
- Two External Clock modes, Up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator Options allow Microcontroller and USB module to Run at Different Clock Speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
  - Compare is 16-bit, max. resolution 83.3 ns (TCY)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

- C Compiler Optimized Architecture with Optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: > 40 Years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Optional Dedicated ICD/ICSP Port (44-pin, TQFP package only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I²C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

7.0 DATA EEPROM MEMORY

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space, but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire VDD range.

Four SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- EECON1
- EECON2
- EEDATA
- EEADR

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADR register holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer; it will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Table 28-1 in **Section 28.0 "Electrical Characteristics"**) for exact limits.

7.1 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 7-1) is the control register for data and program memory access. Control bit, EEPGD, determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit, CFGS, determines if the access will be to the Configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access Configuration registers. When CFGS is clear, the EEPGD bit selects either Flash program or data EEPROM memory.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WREN bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software; it is cleared in hardware at the completion of the write operation.

Note:	The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See **Section 6.1 "Table Reads and Table Writes"** regarding table reads.

The EECON2 register is not a physical register. It is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

9.0 INTERRUPTS

The PIC18F2455/2550/4455/4550 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 000008h and the low-priority interrupt vector is at 000018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

Each interrupt source has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The "return from interrupt" instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

Note:	Do not use the `MOVFF` instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

9.1 USB Interrupts

Unlike other peripherals, the USB module is capable of generating a wide range of interrupts for many types of events. These include several types of normal communication and status events and several module level error events.

To handle these events, the USB module is equipped with its own interrupt logic. The logic functions in a manner similar to the microcontroller level interrupt funnel, with each interrupt source having separate flag and enable bits. All events are funneled to a single device level interrupt, USBIF (PIR2<5>). Unlike the device level interrupt logic, the individual USB interrupt events cannot be individually assigned their own priority. This is determined at the device level interrupt funnel for all USB events by the USBIP bit.

For additional details on USB interrupt logic, refer to **Section 17.5 "USB Interrupts"**.

19.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

19.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

19.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual Configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

19.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of the SPI are supported. To accomplish communication, typically three pins are used:

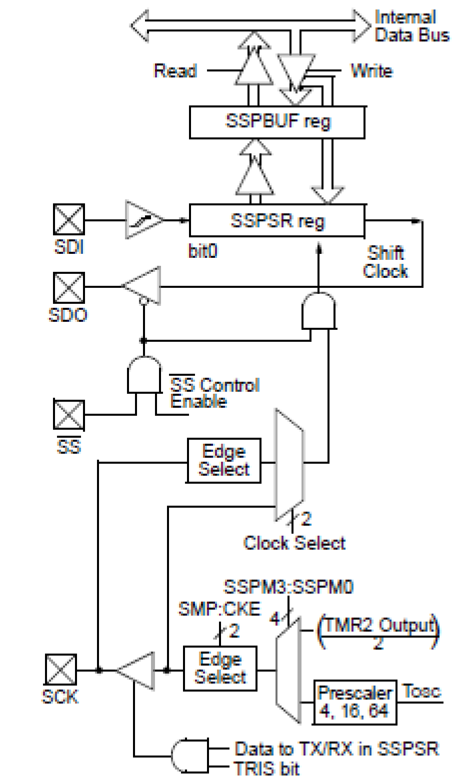
- Serial Data Out (SDO) – RC7/RX/DT/SDO
- Serial Data In (SDI) – RB0/AN12/INT0/FLT0/SDI/SDA
- Serial Clock (SCK) – RB1/AN10/INT1/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) – RA5/AN4/$\overline{SS}$/HLVDIN/C2OUT

Figure 19-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 19-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note: Only those pin functions relevant to SPI operation are shown here.

PROJECT DETAILS

Student Details			
Student Name	**UDAYAN PRABIR SINHA**		
Register Number	110921560	Section / Roll No	A/76
Email Address	udayanprabirsinha@gmail.com	Phone No (M)	9538430294
Student Name	**PRANAY RAJ SINGH**		
Register Number	110921278	Section / Roll No	A/21
Email Address	sng.pranay@gmail.com	Phone No (M)	9008770212
Project Details			
Project Title	**HOME AUTOMATION SYSTEM**		
Project Duration	4 months	Date of reporting	12th January, 2015
Organization Details			
Organization Name	**MANIPAL INSTITUTE OF TECHNOLOGY**		
Full postal address with pin code	Manipal Institute of Technology, Manipal – 576104 (Karnataka State), INDIA		
Website address	http://manipal.edu/mit.html		
Supervisor Details			
Supervisor Name			
Designation			
Full contact address with pin code			
Email address		Phone No (M)	
Internal Guide Details			
Faculty Name	**ANJAN GUDIGAR**		
Full contact address with pin code	Dept of Instrumentation and Control Engg., Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA		
Email address	anjan.gudigar@manipal.edu		