CHAPTER ONE

Introduction

1.1 General

A computer machine like our brain is that whatever you hear or see gets recorded into its hard drive or memory. The human brain is a wonderful complex machine; it even remembers everything from a person's childhood until he/she gets old. When you get older, the memory also is getting older too, and it’s hard to remember everything and also your body getting weaker and weaker. As we get older, most probably we will develop some common diseases like blood pressure, diabetics, cardiac diseases, renal failure, tumors and alzheimer which requires for sure taking on daily bases medicine for life.

The technology revolution give the medical filed the perfect tools to develop new systems based on microcomputers to be able to remind patients about their pills, medications and even to organize food and exercise time.

The EMMS is a computerized medication dispenser for either inpatient or outpatient use. The system is programmed by the patient's healthcare provider and can dispense individual doses of different drugs for up to a month's supply. Multiple EMMS units can be connected to increase the number of medications that can be managed by the system.

Reducing errors in drug identification and dosing can improve patient health. A 2006 Institute of Medicine report estimated that ≥1.5 million patients are harmed each year by medication errors, resulting in a cost of billions of dollars. Many different devices are available to remind patients to take their medication, including pillboxes. However, disorganized and/or random use of these boxes can lead to dosing errors. According to one study, >75% of older adults who used pillboxes did not follow recommended practices.EMMS is

expected to reduce in-home medication management costs and to prevent medication errors and patient no adherence injuries. Because the device self monitors which medications the patient takes and when, the system recognizes when the patient is no adherent to the medication schedule and can notify the patient with a reminder to take the medication[1].

1.2 Problem Statement

- Drug overdoses are usually life threatening and could lead to death sometimes.
- The elderly and alzheimer's patient medication management system is very risky that's why it needs 24 hours from supervision.
- Forgetting about empty drug containers.

1.3. Objectives

1- To interface the servo motor to the microcontroller.
2- To design and implement a solenoid valve driver circuit controlled by the microcontroller.
3- To interface alarm to the microcontroller.
4- To display the time data and pills or exercise time to the patient on Liquid crystal display (LCD).
5- To design and produce a Print circuit board (PCB) that operates the system.
6- Program the microcontroller to compare the time of drug, move container motor, start alarm and display into LCD.
7- Interfacing circuit with circuit board to broadcast pills situation to the pharmacy and nurse workstations.
8- Implementing software to monitor the situation of the pills and automatically prepare a request to refill, this request includes device name and room number.

1.4. Methodology

The thesis methodology is undertakes as follow:

1. Study the health care management.
2. Study microcontroller and other main component.
3. Construct the complete simulation block diagram using proteus software.
4. Evaluate the performance of the system under different operation.

Condition based on simulations result.

1.5 Thesis Layout

This thesis is presented in five chapters. Chapter one gives an introduction to the thesis, including general, problem statement, objectives and methodology. Chapter tow presents previous work, microcontroller, LCD, stepper motor, and other main component, chapter three describe the implementation of the main circuit, chapter four illustrated simulation result and discussion. Finally chapter five provides conclusion and recommendation

CHAPTER TWO

LITERATURE REVIEW AND MAIN COMPONENTS

2.1 Previous works

In hospitals nurse stations had a chart for all the patients they served, and in these charts they had a detailed descriptive form that includes the drugs proscribed for the patient. The dose and time of each dose, and it was their duty to follow up and update the chart to make sure patients get their medicine in time. However, the problem accrued with outpatients, since they do not remember to take the drugs in time, which is a vital factor in treatment.

So, professionals tried to address the issue to be able to support the outpatients and help them to take their medication in time. They came up with several ideas as explained below:

This paper is about dose time indicator, this invention relates to a device for indicating the time that doses of medicine, tonic, and the like, in pill, capsule and other forms, should be taken according to a prescribed schedule, and more particularly pertains to an indicator of this character which is conspicuously positioned on the bottle containing the doses. One of the objects of this invention is the provision of a simply constructed, compact, and reliable dose time indicator which is arranged on the closure for the bottle in a particularly novel advantageous manner such that, the indicator will not interfere with the normal handling of the closure and will assure that due note will take of the dose taking time indicated thereby. It is another object of this invention to provide an indicator such as described which may be read with ease when viewing the bottle from the side thereof and as well as when looking down thereon. Another object of this invention is to provide an indicator such as described wherein the indicating pointer is effectively held against unintentional or accidental movement from position in which it is set

to indicate a specific time for taking a dose. Further, it is an object of this invention to provide an indicator wherein the bottle closure effects the locking of the pointer in set position. Another object hereof is to provide a dose· time indicator where the operation of closing the bottle causes the closure to clamp the pointer against the bottle and lock it in set position. It is an additional object to provide an indicating device such as described which designed so that the operation of turning is or screwing the closure in place or of removing the closure does not cause the pointer to be moved out of set position as shown in figure 2.1 This invention possesses many other advantages and has other objects which may be made more easily apparent from a consideration of several embodiments of the invention. For this purpose there are shown several forms in the drawings accompanying and forming part of the; present specification. These forms will now be described in detail. Illustrating the general principles of the invention; but it is to be understood that this detailed description is not to be taken in a limiting sense, since the scope of the invention is best defined by the appended claims. · Referring to figure 2.1 to figure 2.6[1].

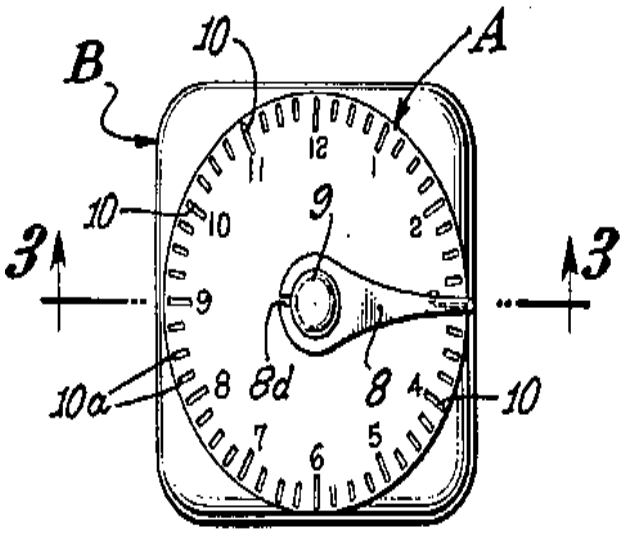


Figure2.1: Top plan view of a dose time indicator embodying the present invention

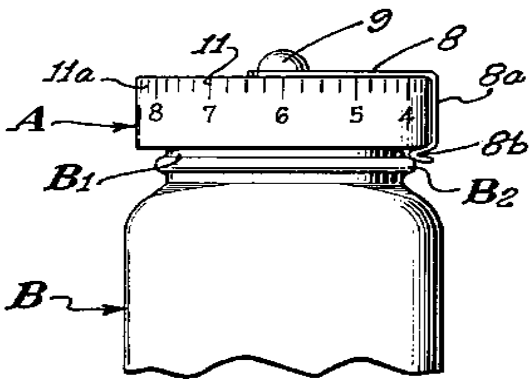


Figure2.2 A side elevation of the indicator

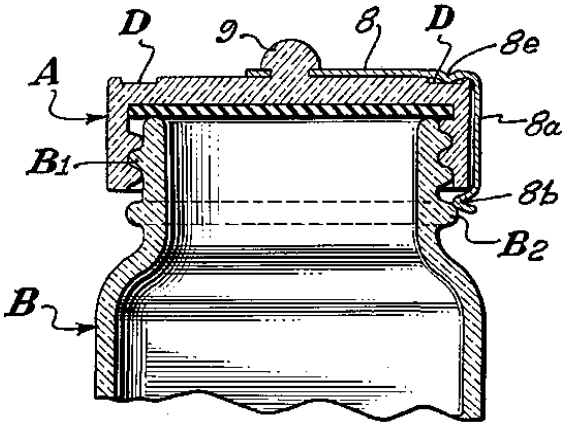


Figure2.3 An enlarged sectional view taken on the line 3-3 of Figure 1,

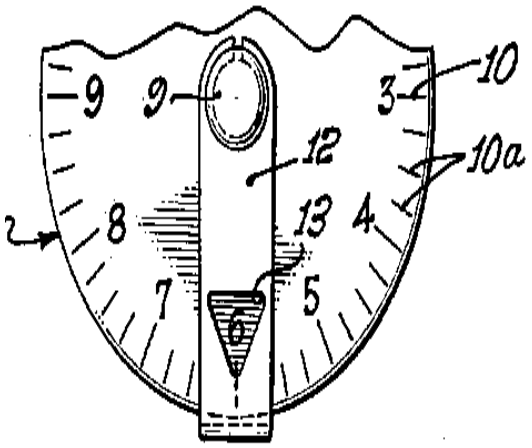


Figure2.4 A fragmentary top plan view of a modified form of this invention

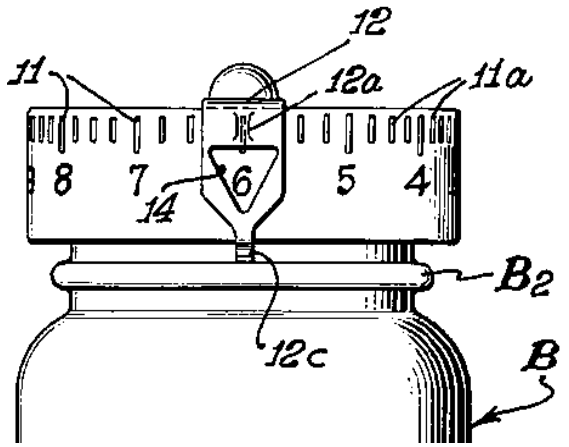


Figure2.5 A side elevation of the indicator

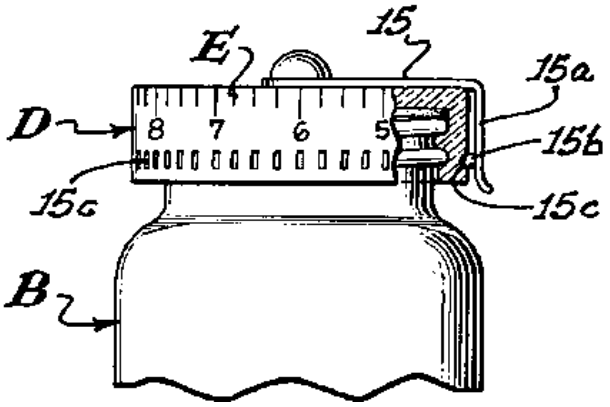


Figure2.6A side elevation, partly in section of another modified form of this invention

The developed a system, is called Dosage Indicating Pill tray, it is described as a tray having individual compartments for holding pills, capsules, or similar solid medication, each compartment being rectangular in plain view and arranged in a rectangular format or seven columns and a plurality of rows. The tray may be loaded with a week's medication for an individual patient with indicia adjacent each column indicating the day of the week, and indicia adjacent the row indicating the time of day that the medication in each compartment is to be taken. A lid or cover cooperates with the wall means defaming the individual Compartments to mutually isolate the compartments

when in the closed position. The inner surfaces of the compartments are preferably rounded in at least one plane of ease of withdrawing medication there from. This system is shown in figures 2.7

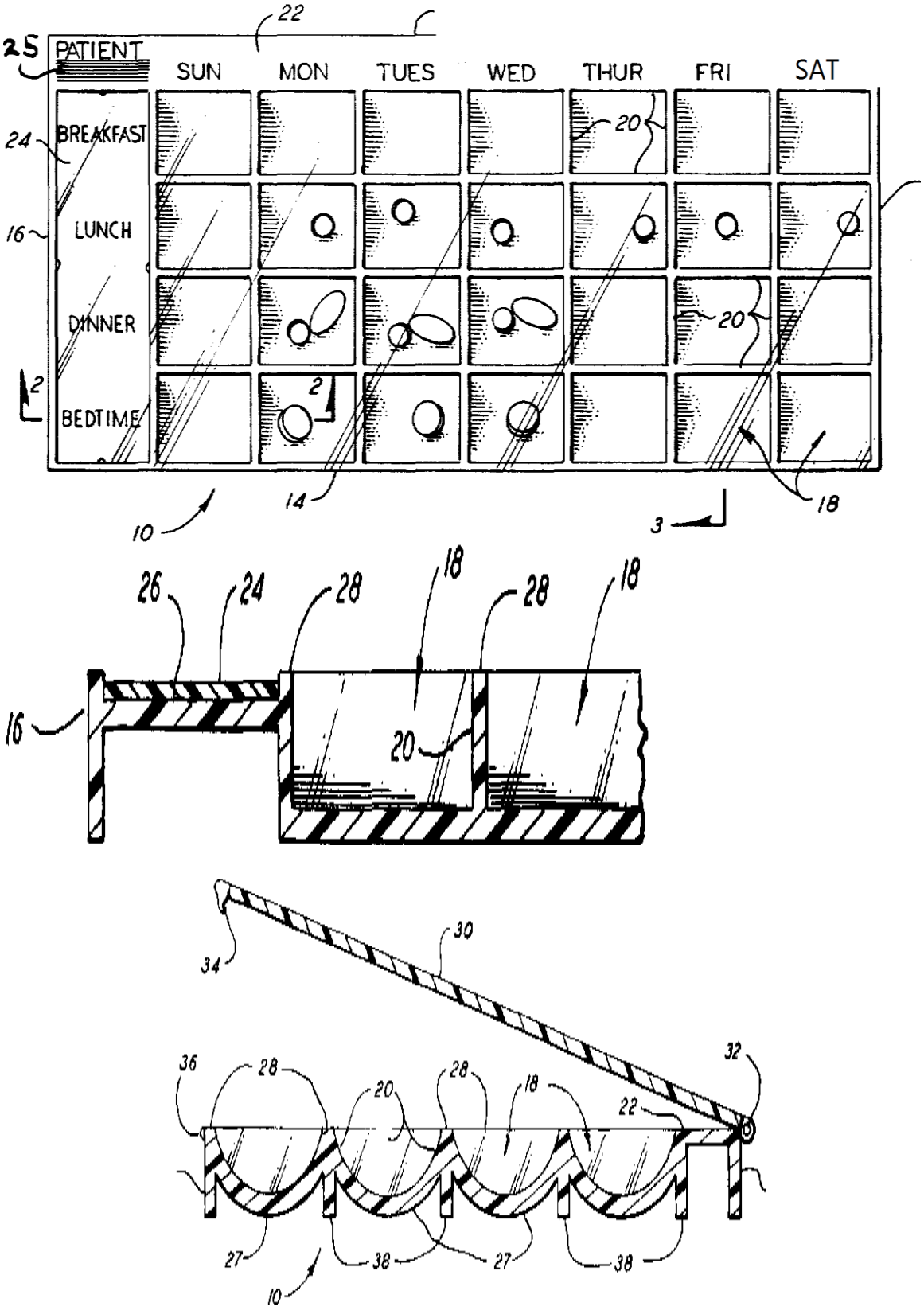


Figure2.7 Dosage Indicating Pill tray

This system is developed called a Medical Reminder Device, which is a reminder device comprising a support on which are located at least two different medicinal substances each of said drugs being in single dose form and an instruction bearing portion on said support adjacent each dose to receive instructions for the use thereof, figure 2.8 shown this system[2].

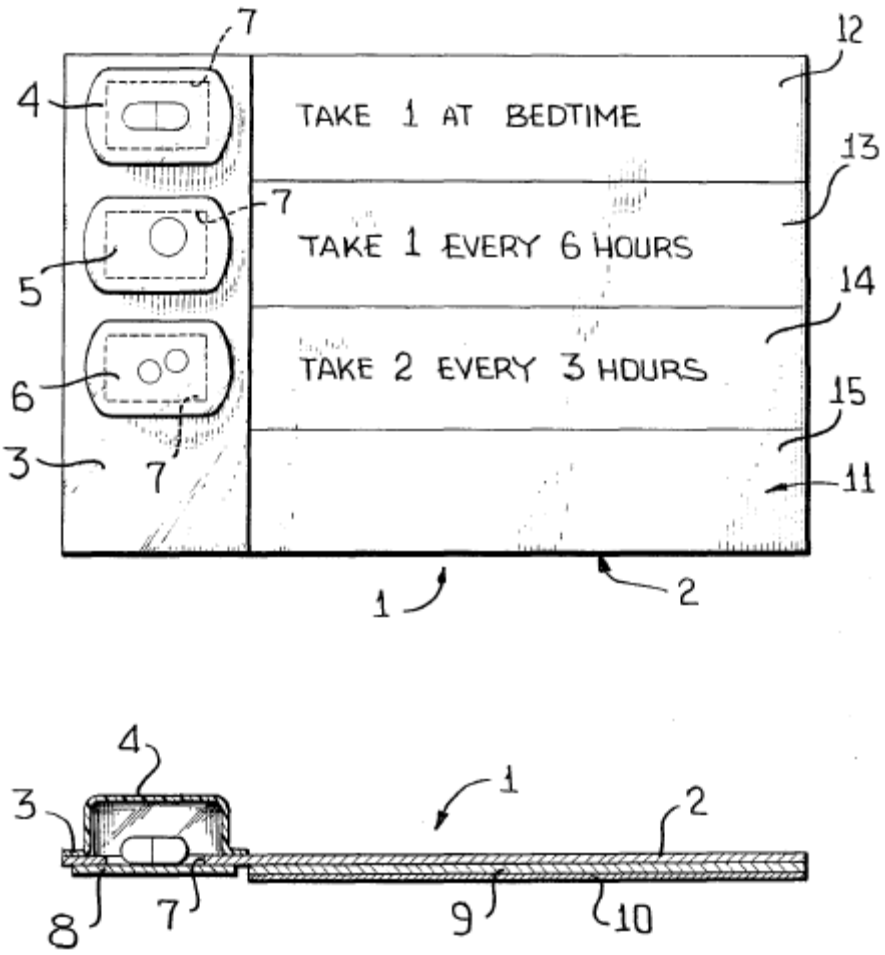


Figure2.8 Medical Reminder Device

A Reminder System for taking medication, it is described as a calendar type dispenser is provided in association with a separable indicia means, the combination comprising a reminder system for reminding users when a particular dosage of medicament must be taken. The medicament container comprises a blister pack dispenser wherein a plurality of blisters is linearly

arranged, in one or more groups, each group having a predetermined number of blisters. Each group has associated therewith a means for retaining a separable overlying planar sheet member provided with a plurality of apertures corresponding to the number of blisters in as elected group. Suitable indicia are marked on the sheet member in associating with each aperture there of thereby identifying the particular dosage in each blister with the time when it should be taken, figure2.9 shown the system [3].

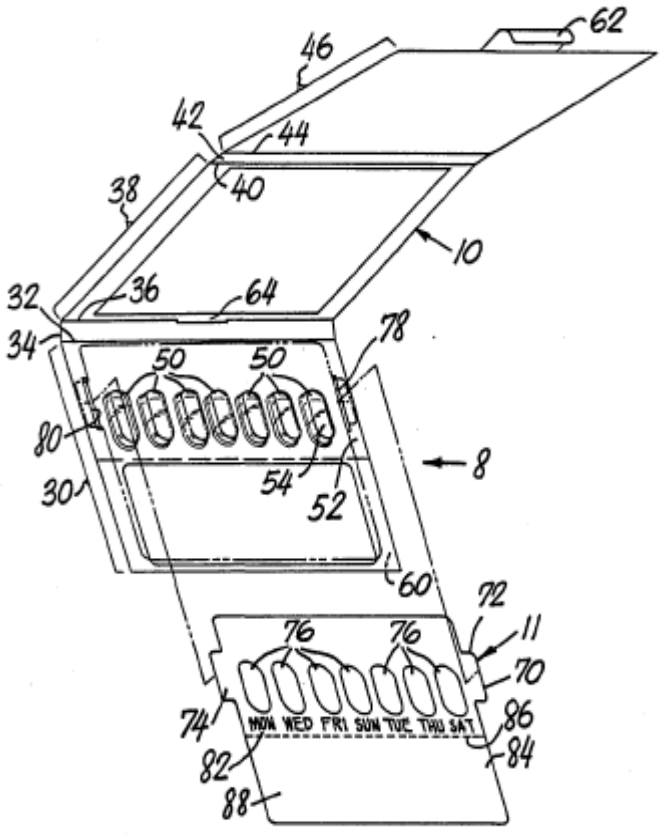


Figure2.9 Reminder System for Taking medication

In this paper different and simpler technology is development in 1995, which is called Reminder Device for Pill Containers, fig 2.10 it is a reminder device with circular dials similar to the face of a clock for pill and medicine containers is adapted for attaching to a pill bottle or other container. The rotatable dials may be manually set to indicators showing the weekday, date and time for taking a pill or other medicinal dose. The reminder device is

disposable and may be pinned to the bottle top or other medicine container or temporarily affixed by means of an adhesive.

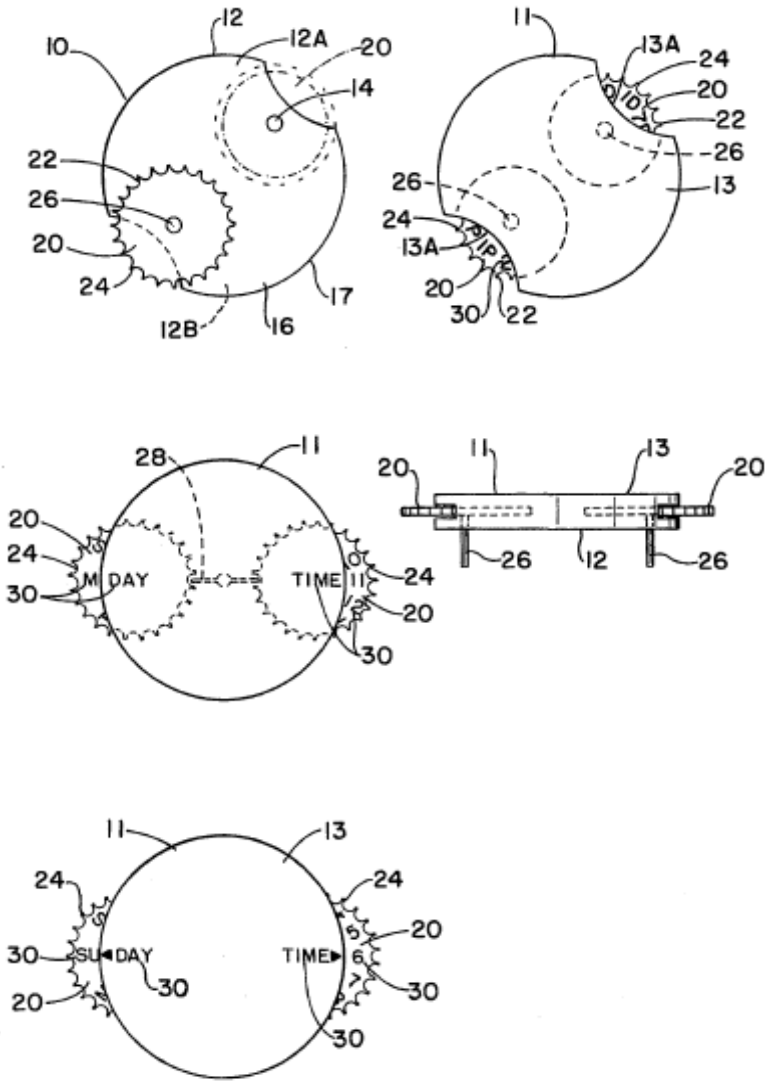


Figure2.7 The Reminder Device for Pill Containers

Another a Dosage Reminder Device and medication carton includes an indicator/support formed from the carton's flat exterior wall retentively engaging a rotatable ring the ring includes dosage time period indicia establishing a dosage schedule and the indicator includes a co-operating next dose pointer for selecting the next dosage time period s indicated. A patient can rotate the ring to align the next scheduled dosage time period with the indicator pointer. The indicator pointer and the scheduled dosage time period

form a reminder indicating when the next dose is due or when the last dose was taken. Figure 2.11shown this system

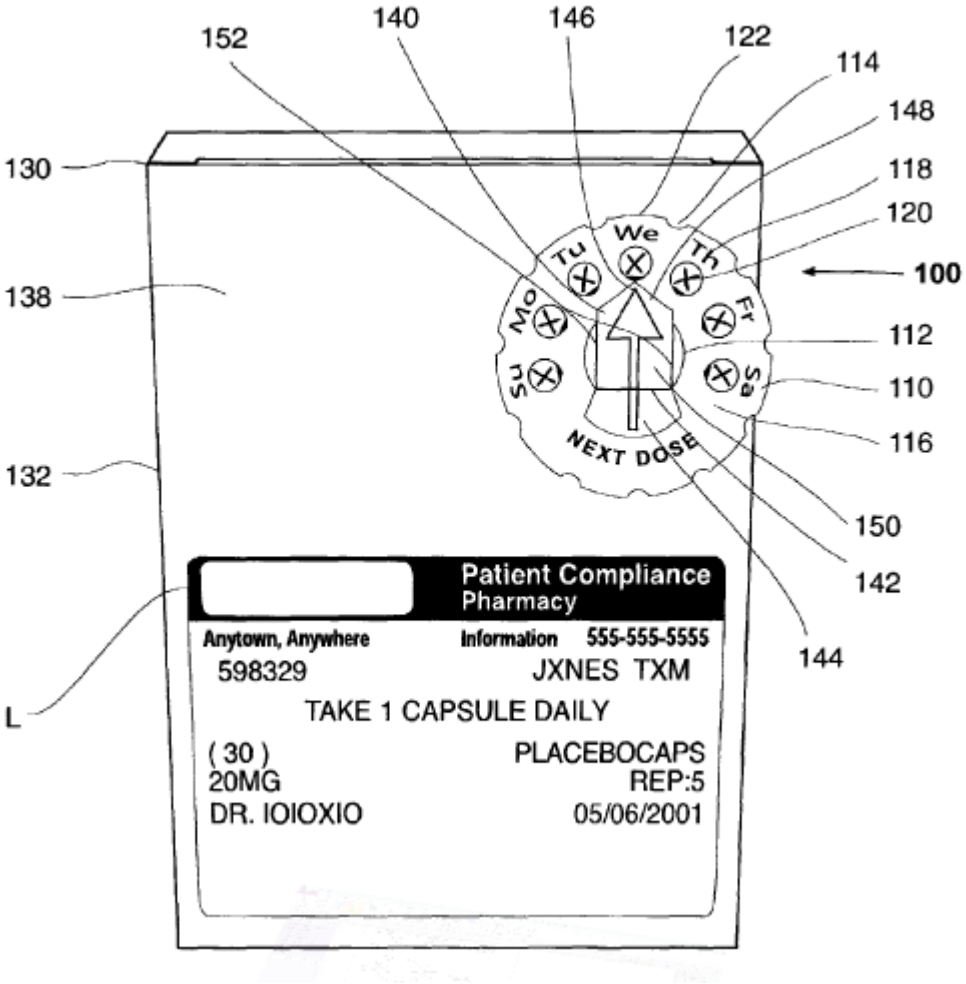


Figure2.11 Dosage Reminder Device and medication carton

In 2006 a Medication Reminder System, which is a cap or closure for medication bottles, containers or vials has a base with a peripheral wall to define a recess. A cover is notably located in the recess below the rim of the peripheral wall. The cover has a marker that is selectively aligned with indicating dosage intervals. A one-way mechanism inhibits rotation of the cover relative to the base in one direction, providing the consumer with a simple mechanical device that acts as a reminder of when the next does in a course of medication is due or when the last dose was taken. The closure permits a single cap to serve the Needs of different medication frequencies

while being inexpensive enough to be included with each container provided by a pharmacy or manufacturer of non-prescription medications or homeopathic products. Show figure 2.12[4]

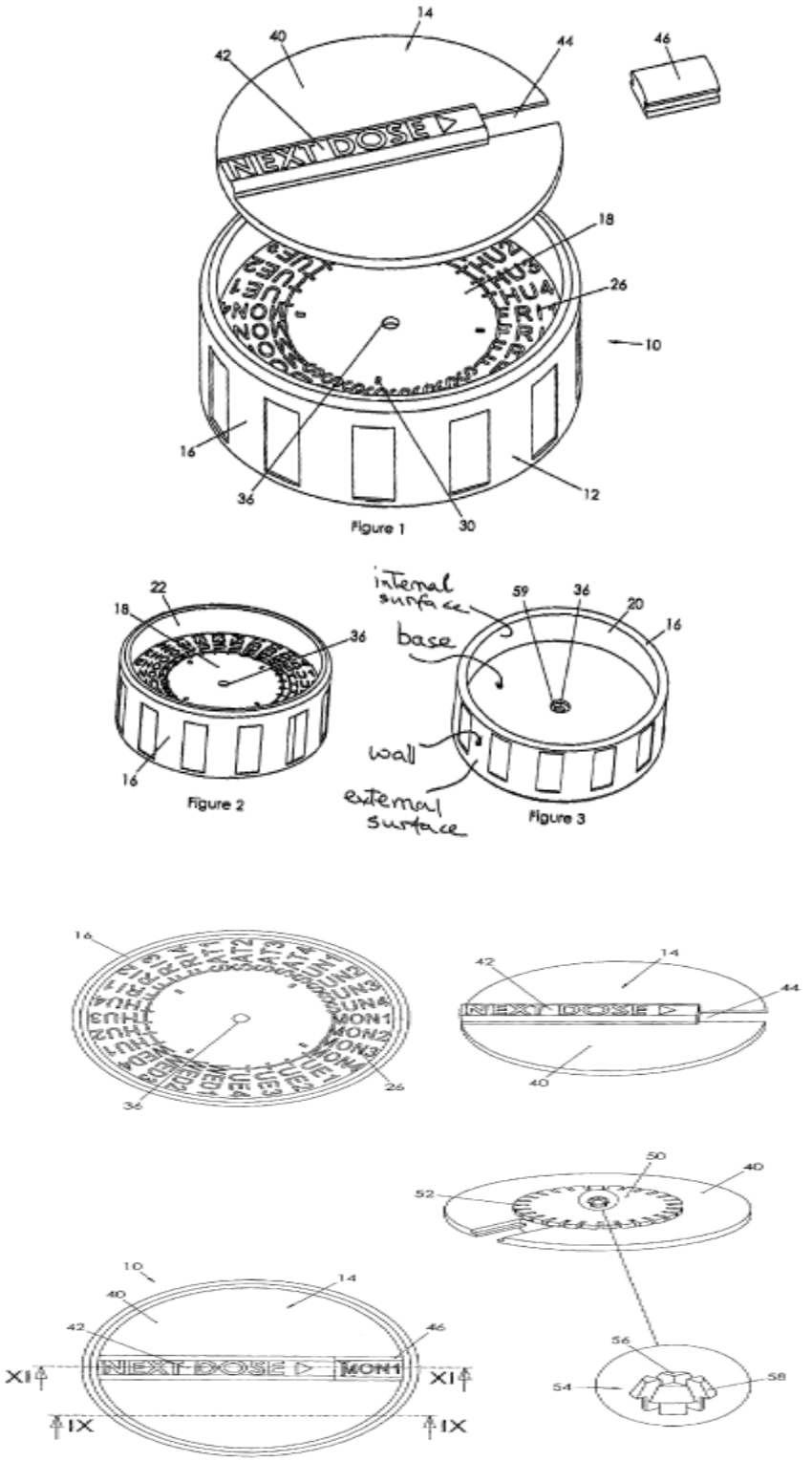


Figure2.8 Medication Reminder System

2.2 Microcontroller

A microcontroller (also MCU or µC) is a functional computer system on a single chip, it contains a processor core, memory, and programmable input/output peripherals. Microcontrollers include an integrated central processing unit (CPU), memory, a small amount of random access memory (RAM), program memory or both and peripherals capable of input and output. Figure 2.13 shows the microcontroller architecture.

Figure2.9 microcontroller architecture

It emphasizes high integration, in contrast to a microprocessor which only contains a CPU the kind used in a personal computer (PC). In addition to the usual arithmetic and logic elements of a general purpose microprocessor, the microcontroller integrates additional elements such as read-write memory for data storage, read only memory (ROM) for program storage, flash memory for permanent data storage, peripherals, and input/output interfaces. At clock speeds of as little as 32 KHz, microcontrollers often operate at very low speed compared to microprocessors, but this is adequate for typical applications. They consume relatively little power (mill watts or even microwatts), and will

generally have the ability to retain functionality while waiting for an event such as a button press or interrupt. Power consumption while sleeping (CPU clock and peripherals disabled) may be just nano watts, making them ideal for low power and long lasting battery applications. Microcontrollers are used in automatically controlled products and devices such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size, cost, and power consumption compared to a design using a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to electronically control many more processes [13].

2.2.1 Programs

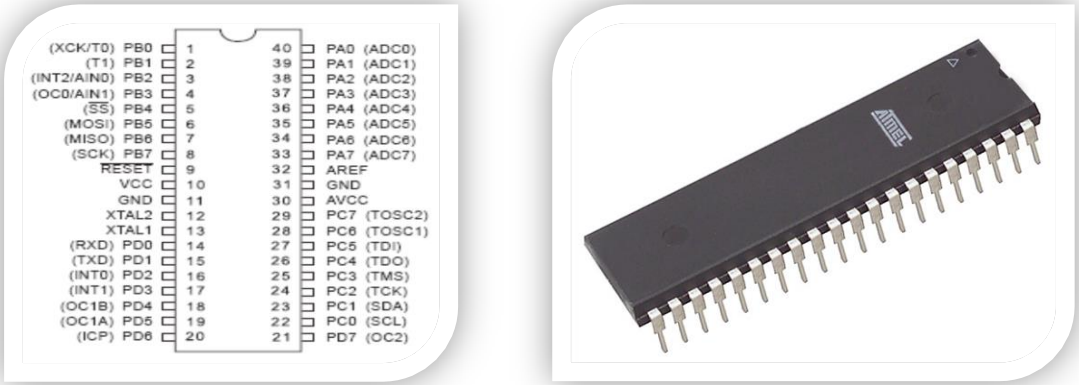
Microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assembly language are used to turn high-level language programs into a compact machine code for storage in the microcontroller's memory. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory, or program memory may be field-alterable flash or erasable read-only memory. Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target microcontrollers. These languages are either designed especially for the purpose, or versions of general purpose languages such as the C programming language. Compilers for general purpose languages will typically have some restrictions as well as enhancements to better support the unique characteristics of microcontrollers. Some microcontrollers have environments to aid developing certain types of applications. Microcontroller vendors often make tools freely available to make it easier to adopt their hardware. Many microcontrollers are so quirky that they effectively require their own non-standard dialects of C, such as SDCC for the 8051, which

prevent using standard tools (such as code libraries or static analysis tools) even for code unrelated to hardware features. Interpreters are often used to hide such low level quirks. Interpreter firmware is also available for some microcontrollers. Simulators are available for some microcontrollers, such as in Microchip's MPLAB environment. These allow a developer to analyze what the behavior of the microcontroller and their program should be if they were using the actual part. A simulator shows the internal processor state and also that of the outputs, as well as allowing input signals to be generated. While on the one hand most simulators will be limited from being unable to simulate much other hardware in a system, they can exercise conditions that may otherwise be hard to reproduce at will in the physical implementation, and can be the quickest way to debug and analyze problems [14].

2.2.2 Atmel AVR

The AVR is a modified harvard architecture 8-bit RISC single chip microcontroller which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one time programmable ROM, Erasable program read only memory(EPROM), or Electrical Erasable program read only memory EEPROM used by other microcontrollers at the time. The original AVR MCU was developed at a local ASIC house in Trondheim Norway, where the two founders of Atmel Norway were working as students. It was known as a μRISC (Micro RISC). When the technology was sold to Atmel, the internal architecture was further developed by Alf and vegard at Atmel Norway, a subsidiary of Atmel founded by the two architects. The acronym AVR has been reported to stand for advanced virtual RISC, but it has also been rumored to stand for the initials of the chip's designers: Alf and Vegard's RISC. Atmel says that the name AVR is not an acronym and does not stand for anything in particular. Note that the use of "AVR" in this article generally refers to the 8-bit RISC line of Atmel AVR microcontrollers. Among the first

of the AVR line was the AT90S8515, which in a 40-pin DIP package has the same pin out as an 8051 microcontroller, including the external multiplexed address and data bus. The polarity of the RESET line was opposite (8051's having an active-high RESET, while the AVR has an active-low RESET), but other than that, the pin out was identical. Figure 2.14 Show the AVR Atmega 8535L microcontroller overview and pin configuration [13].



(a) Pin configuration (b) Over view

Figure2.10: AVR Atmega8535L microcontroller

The pin description of AVR at mega 8535L microcontroller as follows:

VCC: Digital supply voltage.

GND: Ground.

Port A (PA7..PA0): Port A serves as the analog inputs to digital output (A/D) Converter. Also PORTA serves as an 8-bit bi-directional input and output (I/O) port, if the A/D Converter is not used.

Port B (PB7..PB0): Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

Port C (PC7..PC0): Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

Port D (PD7...PD0): Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

RESET: Reset input. a low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running.

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2: Output from the inverting oscillator amplifier.

AVCC: AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the A/D Converter is not used. If the A/D Converter is used, it should be connected to VCC through a low-pass filter.

AREF: AREF is the analog reference pin for the A/D Converter.

2.3 Darlington transistor

In electronics the Darlington transistor is a semiconductor device which combines two bipolar transistors in tandem (often called a "Darlington pair") in a single device so that the current amplified by the first is amplified further by the second transistor. This gives it high current gain (written β or h_{FE}), and takes up less space than using two discrete transistors in the same configuration. The use of two separate transistors in an actual circuit is still very common, even though integrated packaged devices are available. This configuration was invented by Bell Laboratories engineer Sidney Darlington. The idea of putting two or three transistors on a single chip was patented by him, but not the idea of putting an arbitrary number of transistors, which would have covered all modern integrated circuits. As shown in figure 2.15 A similar transistor configuration using two transistors of opposite type (NPN and PNP) is the Sziklai pair, sometimes called the "complementary Darlington".

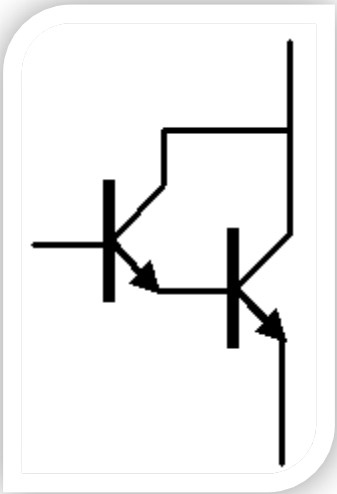


Figure2.15 Darlington transistor

A Darlington pair behaves like a single transistor with a very high current gain. This is beneficial as many commonly-used transistors with high gains have a low current threshold. The total gain of the Darlington is the product of the gains of the individual transistors:

$$\beta_{\text{Darlington}} = \beta_1 \times \beta_2 \qquad (2.1)$$

A typical modern device has a current gain of 1000 or more, so that only a tiny base current is required to make the pair switch on. Integrated devices have three leads (B, C and E) which are equivalent to the leads of a standard individual transistor. The base-emitter voltage is also higher; it is the sum of both base-emitter voltages:

$$V_{BE} = V_{BE1} + V_{BE2} \qquad \textbf{(2.2)}$$

To turn on there must be ~0.6 V across both base-emitter junctions which are connected in series inside the Darlington pair. It therefore requires more than 1.2V to turn on. When a Darlington pair is fully conducting, there is a residual saturation voltage of 0.6V in this configuration, which can lead to substantial power dissipation. Another drawback is that the switching speed can be slow, due to the inability of the first transistor to actively inhibit the current into the

base of the second device. This can make the pair slow to switch off. To alleviate this, a resistor of a few hundred ohms between the second device's base and emitter is often used. Integrated Darlington pairs often include this resistor. It has more phase shift at high frequencies than a single transistor and hence can become unstable with negative feedback much more easily [7].

2.4 Liquid Crystal Display

(LCD) is an electro-optical amplitude modulator realized as a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power.

A comprehensive classification of the various types and electro-optical modes of LCDs is provided in the article LCD classification. Shown figure 2.16[11].

Figure2.11 Liquid crystal display (LCD)

2.5 A stepper Motor

A stepper motor was used as the main motor in this thesis. It rotates by 1.8 degrees with each 15V signal sent to it. This motor was used for precision since it rotates on such small degrees. Thus, the motor was beneficial since it was used for medicine purposes. A stepper motor (or step motor) is a brushless, synchronous electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely without any feedback mechanism (see open-loop controller), as long as the motor is carefully sized to the application. Stepper motors are similar to

switched reluctance motors, which are very large stepping motors with a reduced pole count, and generally are closed-loop commutated.

2.5.1 Fundamentals of operation

Stepper motors operate differently from direct current (DC) brush motors, which rotate when voltage is applied to their terminals. Stepper motors, on the other hand, effectively have multiple toothed electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, such as a microcontroller. To make the motor shaft turn, first one electromagnet is given power, which makes the gear's teeth magnetically attracted to the electromagnet's teeth. When the gear's teeth are thus aligned to the first electromagnet, they are slightly offset from the next electromagnet. So when the next electromagnet is turned on and the first is turned off , the gear rotates slightly to align with the next one, and from there the process is repeated. Each of those slight rotations is called a step,with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle.

2.5.2 Interfacing PIC with stepper motor

For applications where precise measuring of a motors' rotor position is critical, a stepper motor is the best choice. Stepper motors operate differently from other motors; rather than voltage being applied and the rotor spinning smoothly, stepper motors turn on a series of electrical pulses to the motor's windings. Each pulse rotates the rotor by an exact degree. These pulses are called steps, hence the name stepper motor. The degrees per pulse are set in the motor’s manufacturing, and are provided in the spec sheets for that motor. They can range from ultra-fine movements of a fraction of a degree (i.e., 0.10 degrees) to larger steps (i.e. 62.5 degrees).

This article will explain the operating principals of stepper motors, and will give instructions to how control them via a PIC16F84 microcontroller to perform many functions. Stepper motors consist of a permanent magnet rotating shaft, called the rotor, and electromagnets on the stationary portion that surrounds the motor, called the stator. Figure 2.17 illustrates one complete rotation of a stepper motor. At position 1, we can see that the rotor is beginning at the upper electromagnet, which is currently active (has voltage applied to it). To move the rotor Clock Wise (CW), the upper electromagnet is deactivated and the right electromagnet is activated, causing the rotor to move 90 degrees CW, aligning itself with the active magnet. This process is repeated in the same manner at the south and west electromagnets until we once again reach the starting position.

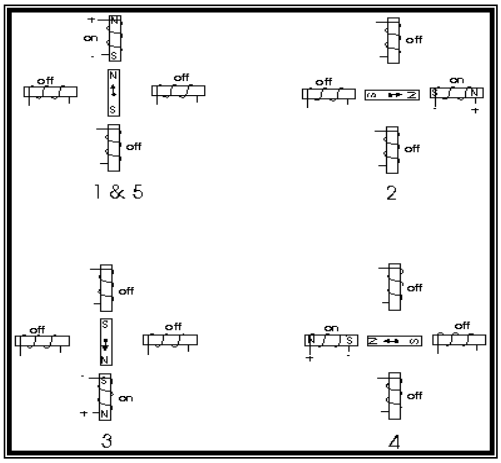


Figure2.17 Starting position of motor

In the above example, we used a motor with a resolution of 90 degrees or demonstration purposes. In reality, this would not be a very practical motor for most applications. The average stepper motor's resolution -- the amount of degrees rotated per pulse -- is much higher than this. For example, a motor with a resolution of 5 degrees would move its rotor 5 degrees per step, thereby requiring 72 pulses (steps) to complete a full 360 degree rotation.

You may double the resolution of some motors by a process known as "half-stepping". Instead of switching the next electromagnet in the rotation on one

at a time, with half stepping you turn on both electromagnets, causing an equal attraction between, thereby doubling the resolution. As you can see in Figure 2.18 in the first position only the upper electromagnet is active, and the rotor is drawn completely to it. In position 2, both the top and right electromagnets are active, causing the rotor to position itself between the two active poles. Finally, in position 3, the top magnet is deactivated and the rotor is drawn all the way right. This process can then be repeated for the entire rotation.

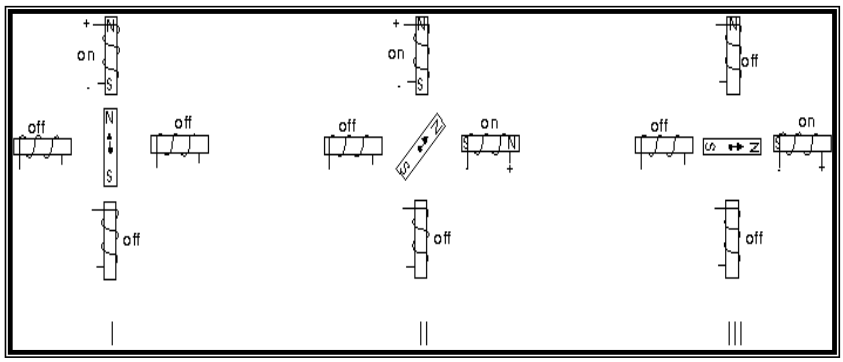


Figure2.18 The rotor position

There are several types of stepper motors. 4-wire stepper motors contain only two electromagnets; however the operation is more complicated than those with three or four magnets, because the driving circuit must be able to reverse the current after each step. For our purposes, we will be using a 6-wire motor.

Unlike our example motors which rotated 90 degrees per step, real-world motors employ a series of mini-poles on the stator and rotor to increase resolution. Although this may seem to add more complexity to the process of driving the motors, the operation is identical to the simple 90 degree motor we used in our example. An example of a multi pole motor can be seen in Figure 2.19 In position 1, the north pole of the rotor's permanent magnet is aligned with the south pole of the stator's electromagnet. Note that multiple positions are aligned at once. In position 2, the upper electromagnet is deactivated and the next one to its immediate left is activated, causing the rotor to rotate a

precise amount of degrees. In this example, after eight steps the sequence repeats [9].

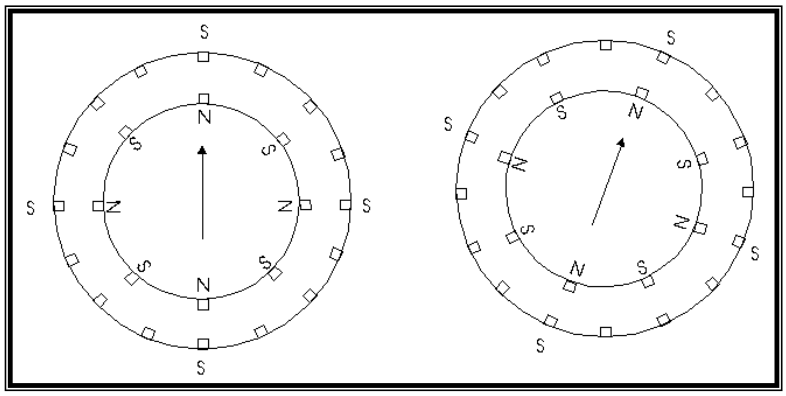


Figure2.19 Multi pole motor

The specific stepper motor we are using for our experiments (ST-02: 5VDC, 5 degrees per step) has 6 wires coming out of the casing. If we follow Figure 2.20 the electrical equivalent of the stepper motor, we can see that 3 wires go to each half of the coils, and that the coil windings are connected in pairs. This is true for all four-phase stepper motors.

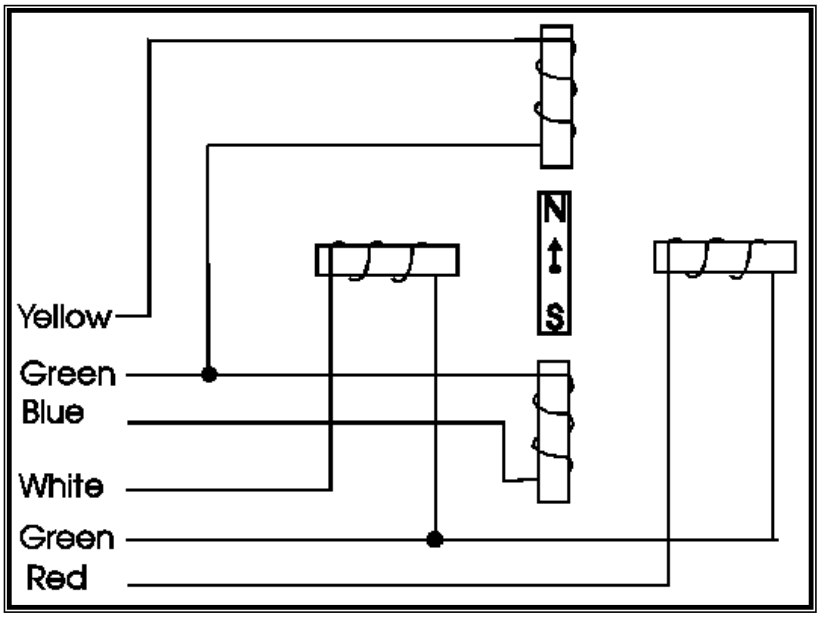


Figure2.20 The specific steeper motor

However, if you do not have an equivalent diagram for the motor you want to use, you can make a resistance chart to decipher the mystery connections.

There is a 13 ohm resistance between the center-tap wire and each end lead, and 26 ohms between the two end leads. Wires originating from separate coils are not connected, and therefore would not read on the ohm meter.

2.6 A buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or electronic. Typical uses of buzzers and beepers include alarms, timers and confirmation of user input such as a mouse click or keystroke. Shows figure 2.21[5]

Figure2.21 A buzzer

2.7 Solenoid Valve

A solenoid valve is an electromechanical valve for use with liquid or gas. The valve is controlled by an electric current through a solenoid: in the case of a two-port valve the flow is switched on or off; in the case of a three-port valve, the outflow is switched between the two outlet ports. Multiple solenoid valves can be placed together on a manifold. Solenoid valves are the most frequently used control elements in fluidics. Their tasks are to shut off, release, dose, distribute or mix fluids. They are found in many application areas. Solenoids offer fast and safe switching, high reliability, long service life, good medium compatibility of the materials used, low control power and compact design.

Besides the plunger-type actuator which is used most frequently, pivoted-armature actuators and rocker actuators are also used. A solenoid valve has two main parts the solenoid and the valve. The solenoid converts electrical energy into mechanical energy which, in turn opens or closes the valve

mechanically. A direct acting valve has only a small flow circuit, shown within section E of this diagram (this section is mentioned below as a pilot valve). This diaphragm piloted valve multiplies this small flow by using it to control the flow through a much larger orifice. Shows figure 2.22

Figure2.22 Solenoid valves

2.8 Relay

A relay is an electrical switch that opens and closes under the control of another electrical circuit. In the original form, the switch is operated by an electromagnet to open or close one or many sets of contacts. Because a relay is able to control an output circuit of higher power than the input circuit, it can be considered, in a broad sense, to be a form of an electrical amplifier. in figure 2.23 shows the relay[9]

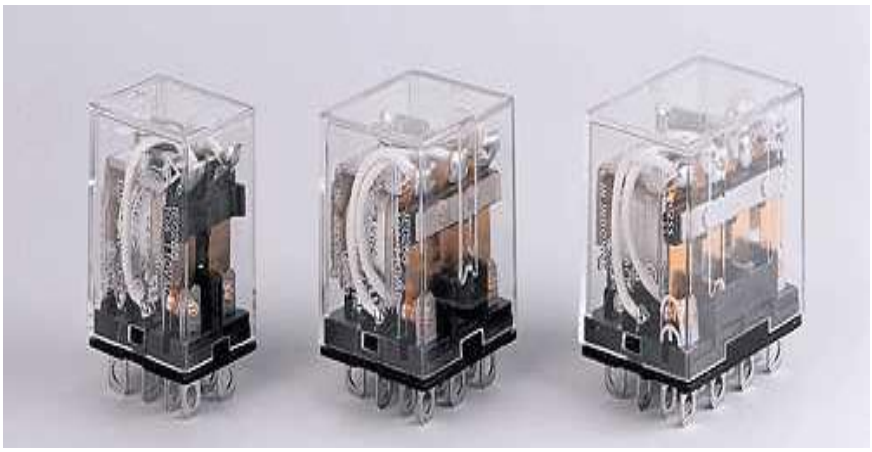


Figure2.23 Relay

2.9 Resistors

A resistor is a two-terminal electronic component designed to oppose an electric current by producing a voltage drop between its terminals in proportion to the current. That is in accordance with Ohm's law V = IR. The resistance R is equals to the voltage drop V across the resistor divided by the current I through the resistor. figure 2.24 show the resistance

Figure2.24 Resistors

2.10Voltage Regulator

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. A voltage regulator may be a simple feed forward. Design or may include negative feedback control loops. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more Alternating current (AC) or (DC) voltages.figure2.25 shows the type of voltage regulator

Figure2.25 A popular three pin 12 V DC voltage regulator IC.

Electronic voltage regulators are found in devices such as computer power supplies where they stabilize the DC voltages used by the processor and other elements. In automobile alternators and central power station generator plants, voltage regulators control the output of the plant. In an electric power distribution system, voltage regulators may be installed at a substation or along distribution lines so that all customers receive steady voltage independent of how much power is drawn from the line [6].

2.11 Crystal Oscillator

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits designed around them became known as crystal oscillators. Figure2.26 shows the crystal oscillator.

Figure2.26 Crystal Oscillator

Quartz crystals are manufactured for frequencies from a few tens of kilo hertz to tens of Mega Hertz. More than two billion crystals are manufactured

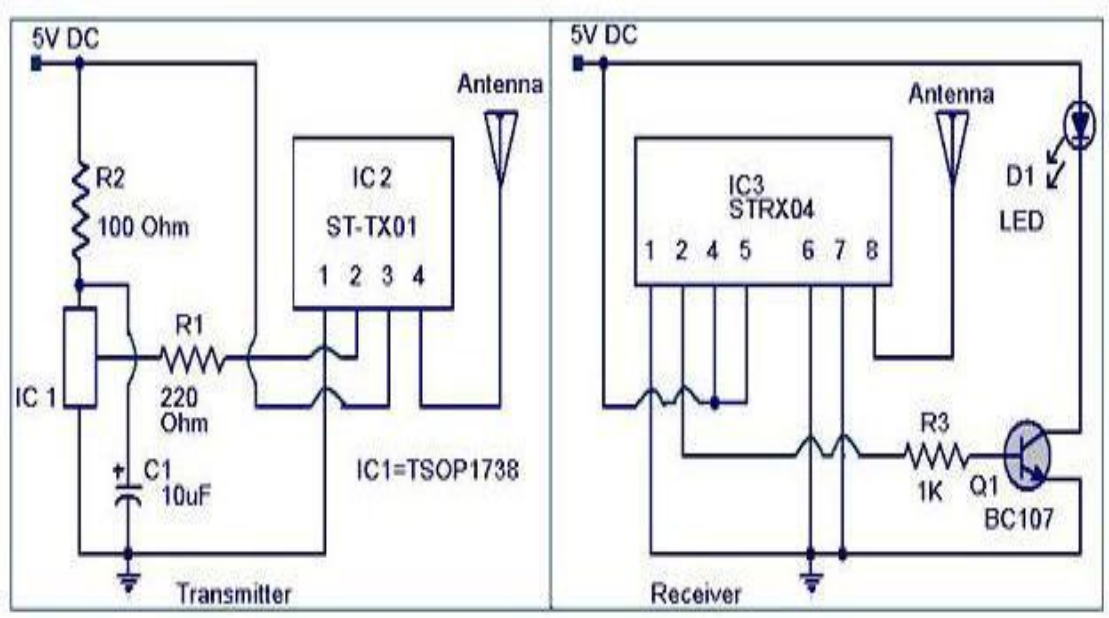
annually. Most are used for consumer devices such as wrist watches, clocks, radios, computers, and cellphones. Quartz crystals are also found inside test and measurement equipment, such as counters, signal generators, and oscilloscopes. [7]

2.12 Capacitor

A capacitor (formerly known as condenser) is a device for storing electric charge. The forms of practical capacitors vary widely, but all contain at least two conductors separated by a non-conductor. Capacitors used as parts of electrical systems, for example, consist of metal foils separated by a layer of insulating film. A capacitor is a passive electronic component consisting of a pair of conductors separated by a dielectric (insulator). When there is a potential difference (voltage) across the conductors, a static electric field develops across the dielectric, causing positive charge to collect on one plate and negative charge on the other plate. Energy is stored in the electrostatic field. An ideal capacitor is characterized by a single constant value, capacitance, measured in farads. This is the ratio of the electric charge on each conductor to the potential difference between them. Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass, in filter networks, for smoothing the output of power supplies, in the resonant circuits that tune radios to particular frequencies and for many other purposes. The capacitance is greatest when there is a narrow separation between large areas of conductor hence capacitor conductors are often called "plates," referring to an early means of construction. In practice the dielectric between the plates passes a small amount of leakage current and also has an electric field strength limit, resulting in a breakdown voltage, while the conductors and leads introduce an undesired inductance and resistance. A ceramic capacitors with 22PF capacitance used in this thesis to complete the crystal oscillator to oscillate to run the internal pointer of the PIC microcontroller [7].

2.13 Modulator and Demodulator

ST-TX01-ASK is designed by the Saw Resonator, with an effective low cost, small size, and simple to use for designing. The main feature of ST-TX01-ASK is Frequency range 315 / 433.92 MHZ and supply voltage: 3~12V figure 2.27 Shows the ST-TX01-ASK Transmitter and recovers.



(a) Transmitter (b) Receiver

Figure2.27 The ST-TX01-ASK

2.14 Encoders and Decoders

The Radio Frequency (RF) spectrum is filled with noise and other signals, especially those frequencies where unlicensed transmitter operation under FCC part 15 rules is allowed. When using a wireless remote control system it is desirable to have a way of filtering out or ignoring those unwanted signals to prevent false data from being received. One way to accomplish this is to use microprocessors at the transmitter and receiver that are programmed with error detection and correction algorithms something like those used in modems. A much simpler way is to use an encoder IC at the transmitter and a decoder IC at the receiver. These ICs automatically generate and decode

multiple serial codes that must match before data is accepted as valid. In the early days of radio control, before these coding ICs were available, radio controlled garage doors sometimes opened themselves when they received transmissions from a plane passing overhead or a two-way radio operating in the area. Encoding and decoding is now used in most wireless control systems to prevent this type of interference and to provide security [5].

2.14.1 Encoder HT12E

The 212 encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12_N data bits. Each address and data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E further enhances the application flexibility of the 212 series of encoder's .figure 2.28 shows HT12E encoder.

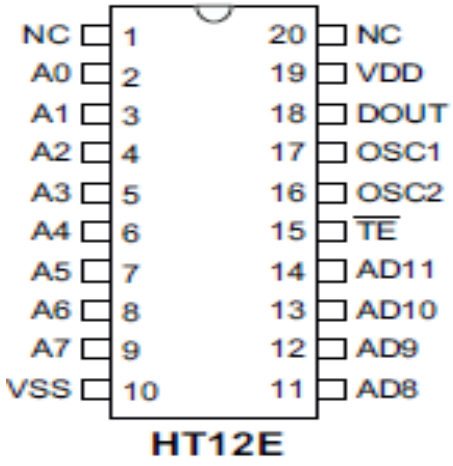


Figure2.28 Encoder HT12E

2.14.2 Decoder HT12D

The 212 decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek_s 212 series of encoders refer to the

encoder and decoder cross reference table. For proper operation, a pair of encoder and decoder with the same number of addresses and data format should be chosen. The decoders receive serial addresses and data from a programmed 212 series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are Decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 212 series of decoders are capable of decoding information that consists of N bits of address and 12_Nbits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits. Figure2.29 shows HT12E decoder13].

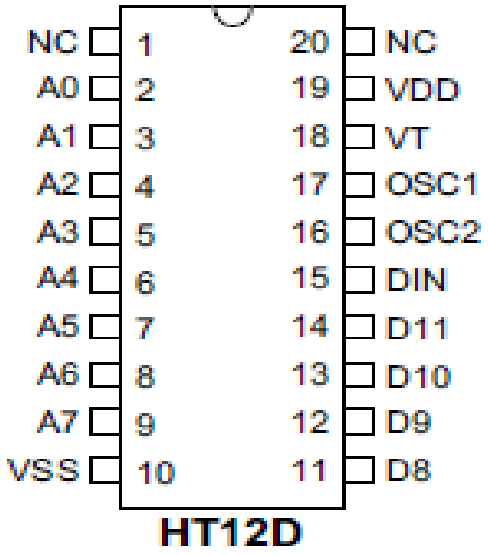


Figure 2.29 Decoder HT12E

2.15 Line Print Terminal

LPT (Line Print Terminal) is the original, and still common, name of the parallel port interface on IBM PC-compatible computers. It was designed to operate a text printer that used IBM's 8-bit extended ASCII character set. The name derives from the fact that "line printer" was a common generic term at the time for any type of text printer figure2.30 shows the LPT pin configuration.

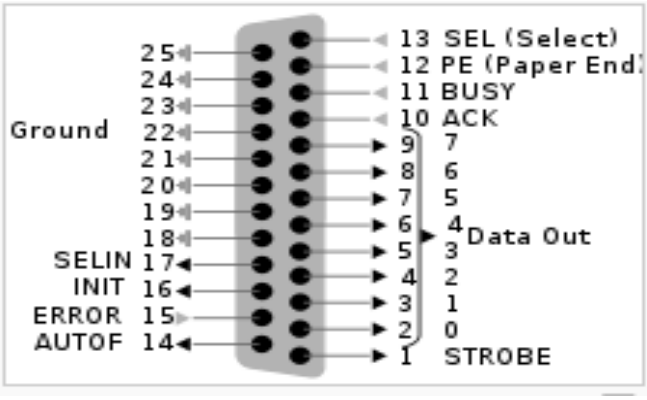


Figure 2.30 LTP Pin Configurations

Graphical printers, along with a host of other devices, have been designed to communicate with the system. It was a de facto industry standard for many years, and was finally standardized as IEEE 1284 in the late 1990s. Today, the parallel port interface is seeing decreasing use because of the rise of Universal Serial Bus (USB) and five wire (IEEE 1394) devices; along with network printing using Ethernet. [7]

CHAPTER THREE

System Implementation

3.1 Introduction

In this chapter the computer model is illustrated along with the details on the program steps inside the microcontroller. Also the circuit diagram simulation is illustrated with a deep analysis.

3.2 System Block Diagram

The system block diagram consists of two main block diagram as follows:

3.2.1 Transmitter block diagram

Figure 3.1 shows the transmitter block diagram. The main parts are:

LCD display: alphanumeric LCD used to display readings.

Switches: used as an input to microcontroller to adjust settings.

Motor drive: integrated Circuit used to drive motors.

Motors: DC motor used to move the cabinets.

Interface circuit: used to interface between low powers to high power circuit.

Encoder: Encodes data to be transmitted.

Transmitter: transmitter module used to modulate data.

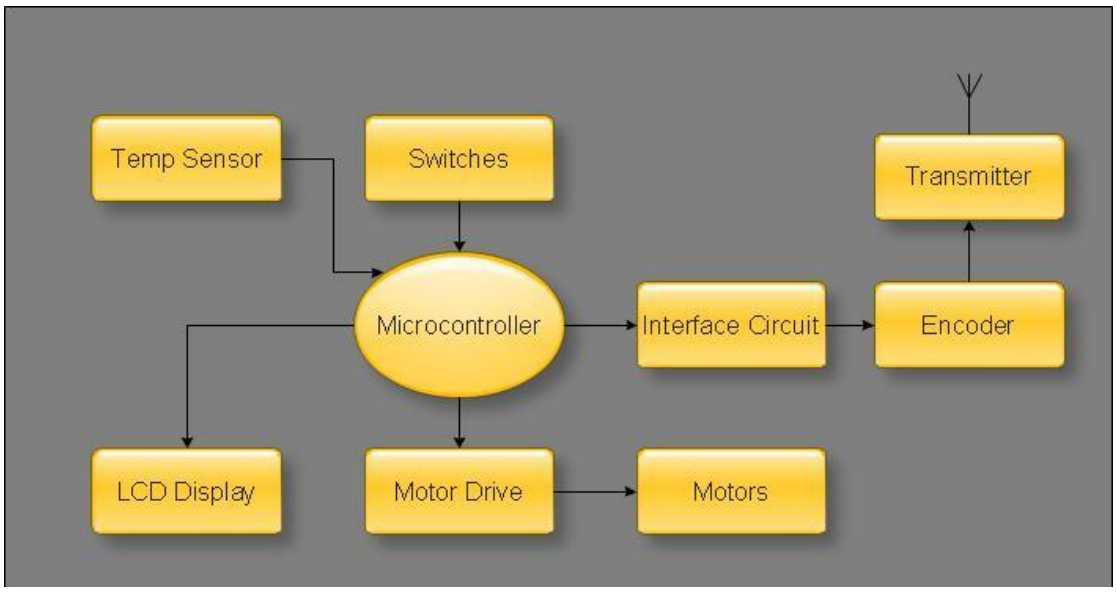


Figure3.1 Transmitter block diagram

3.2.2 Receiver block diagram

Figure 3.2 shows the receiver block diagram. The main parts are:

PC: personal computer.

Receiver: demodulator circuit.

Decoder: decodes received data.

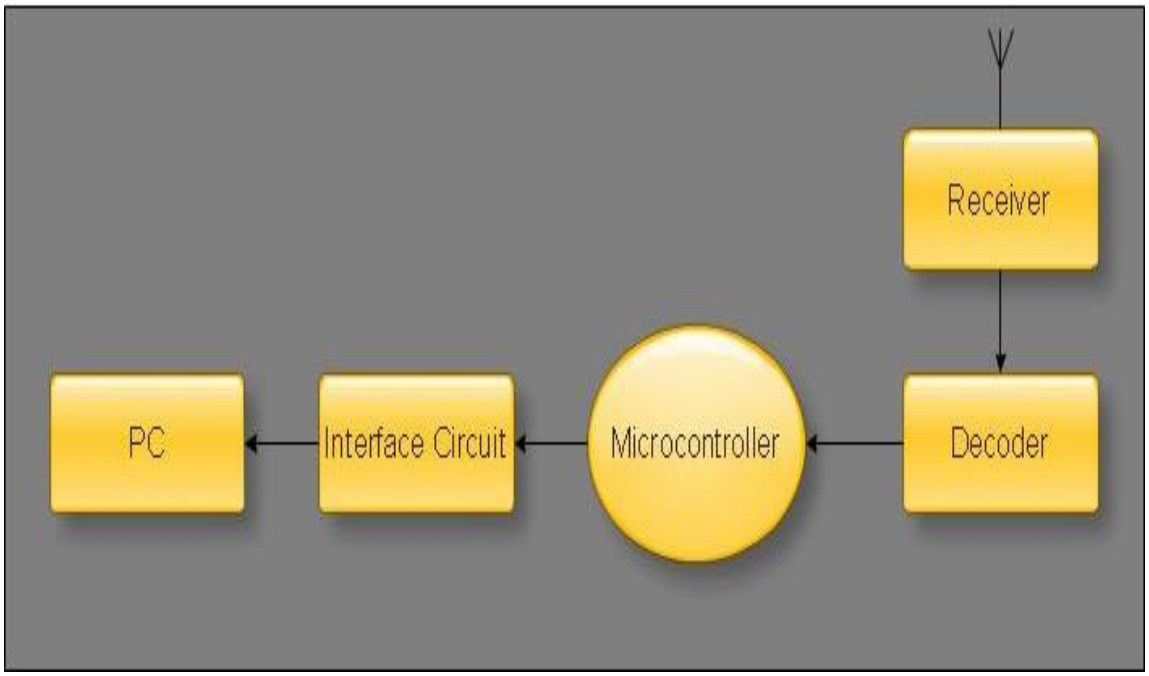


Figure3.2 Receiver block diagram

3.3 flowchart

The flowchart illustrates the data flow of the software inside the microcontroller, as shown in figure 3.3

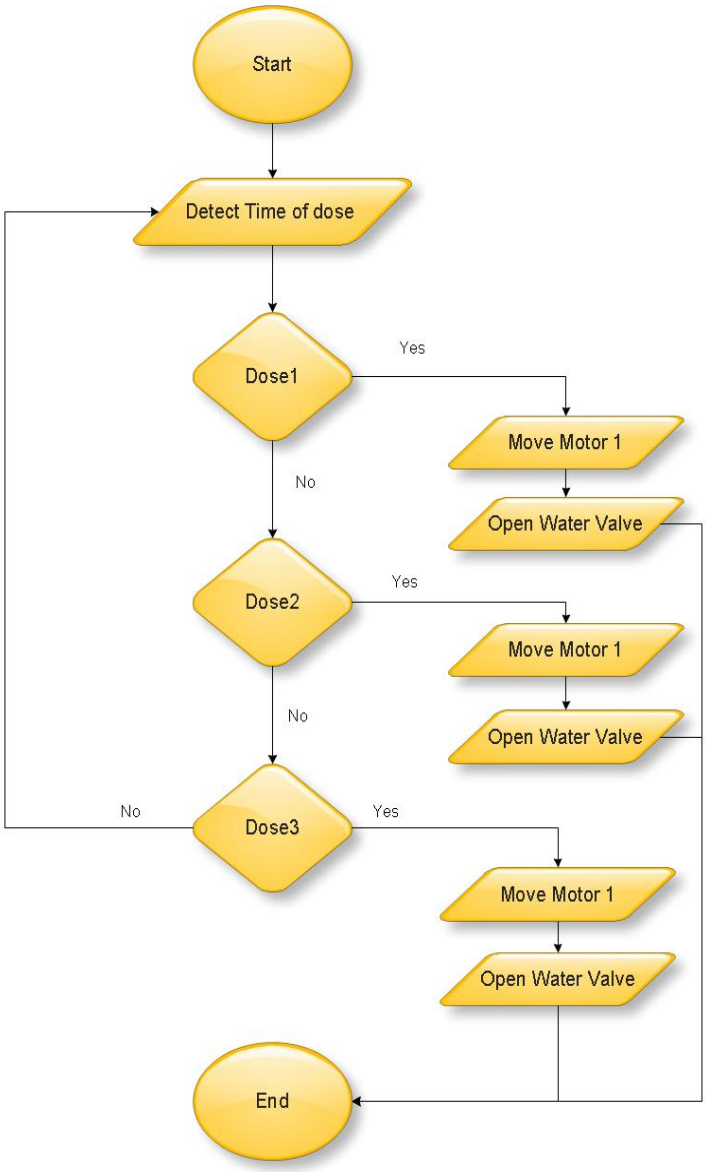


Figure3.3 flow chart

3.4 Transmitter Circuit

The circuit is consists of microcontroller Atmega16 used to control the four containers. The motors interfaced to the microcontroller through motor driver IC that produces a 5 volt, 600 mA. Figure 3.4 shows the transmitter circuit diagram.

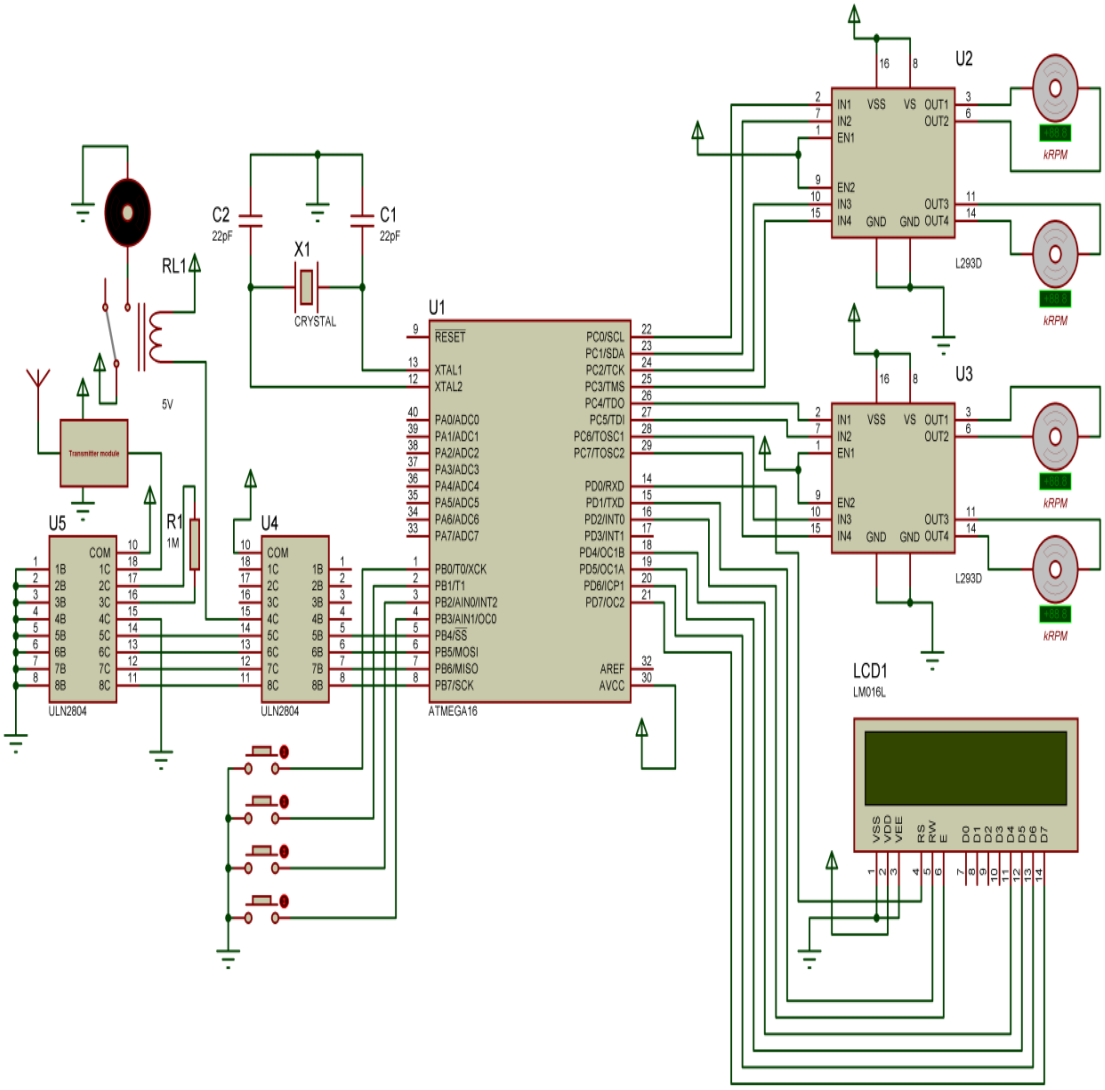


Figure3.4 Transmitter circuit diagram

The main components used to build complete transmitter circuit are:

3.4.1 Microcontroller

The microcontroller Atmega16, U1 is used to control the movement of motors through motor driver. The valve is controlled through relay and current amplifier and displays the status and the process in the LCD display. Moreover, encoder is interfaced to transmit the situation to the receiver circuit remotely, and crystal oscillator is used to speed up the processing of the microcontroller to 800,000 instruction/s. figure3.4 shows Atmel Microcontroller Type Atmega16

3.4.2 Crystal oscillator

An external oscillator with an 8MHz frequency interfaced to the microcontroller PINs 12 and 13 Xtal1 and Xtal2, as shown in figure 3.4 The crystal oscillator connected to ceramic capacitors 22pF according to datasheet of Atmega16.

3.4.3 Liquid crystal display (LCD)

The alphanumeric LCD is used to display the situation inside the microcontroller and the status of the cabinets. It is interfaced directly to microcontroller through Port D (D0 to D7 While D3 is excluded).figure 3.4 shows connected LCD

3.4.4 Switches/ buttons

Four buttons are interfaced to microcontroller in order to adjust the setting of the device including the delay and the interval. And it was connected to PORTB (B.0 to B.3). Shown figure 3.4

3.4.5 IC ULN2804

This integrated circuit was used as a current amplifier since the encoder require a 100mA trigger and the microcontroller can only produce 40mA, U4 is also used to switch on/off relay that is responsible for controlling the valve. Shows figure 3.9

3.4.6 IC L293

Half bridge integrated circuit and known as motor driver, this integrated circuit is used to control high power motors from a control signal from the microcontroller which give an output up to 35 volt and up to 500mA and it

protect the circuit from reverse current that my occur during the motor movement.U2 and U3 shows figure 3.4

3.4.7 Encoder HT12E

HT stands for Hitachi Company and 12 stands or (8 address+4 Data), the integrated circuit converts the output of the ULN2804 that comes from the microcontroller as 4 bit into serial and add the address to be transmitted through modulator.U4 and U5 shows figure 3.4

3.5 Receiver Circuit

The receiver circuit starts with a demodulator which demodulated signal and pass it to the decoder that checks the address of the received signal with the address of the decoder, if it match a four bot output from decoder to the microcontroller is detected. The microcontroller then sends the device status to the LCD for analysis. Figure 3.12 shows the complete receiver circuit diagram

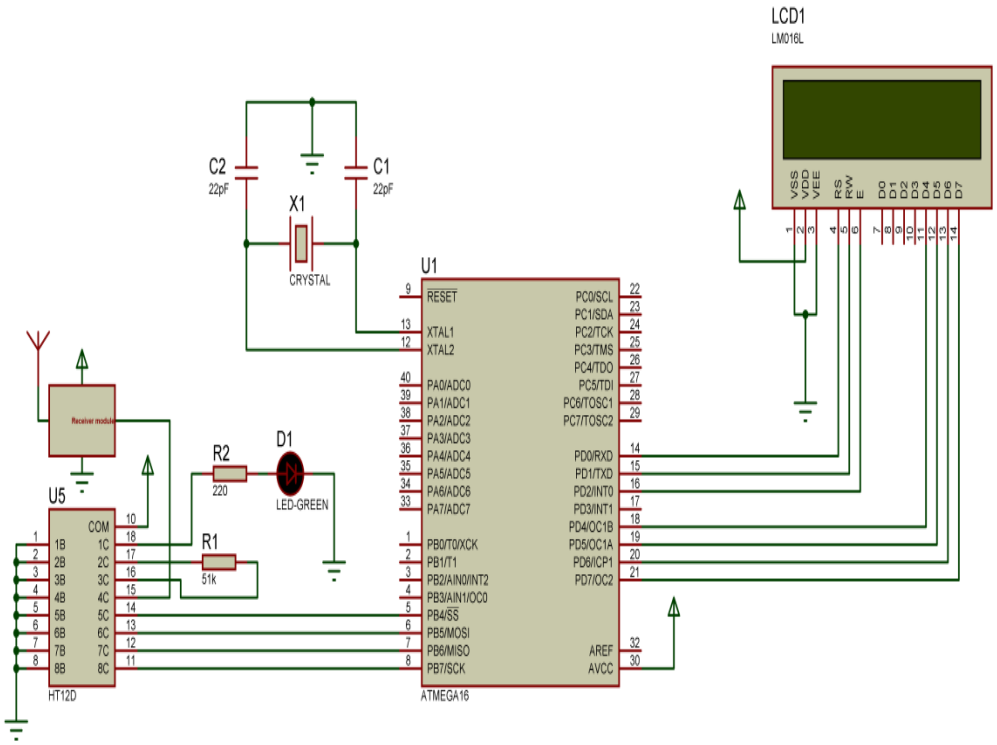


Figure3. 5 Receiver circuit diagram

3.5.1Microcontroller

Used as monitoring to the received signals from the receiver circuit and it passes the status to the personal computer through LPT PORT. As shows in figure3.5

3.5.2Decoder HT12D

HT stands for Hitachi Company and 12 stands or (8 address + 4 Data), The integrated circuit used to receive the data and decode it from the receiver module to the microcontroller and covert the signal from serial to parallel.U5 shows figure 3.5

3.5.3Local Printer Port

Used as input from microcontroller to personal computer and the status pins were used 10,11,12,13 and from 18 to 25 grounded. As shows in figure 3.15

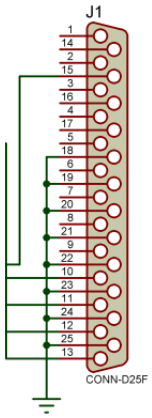


Figure3.6Line printer port 25pin configuration

3.5.4 Regulator

The regulator LM7805 used to drop the voltage of the battery to 5 volt with a maximum of 1.5A current to supply all of the circuit components. Figure 3.7 shows the LM7805 power regulator circuit.

3.5.5 Resistor

Used to drop the voltage before the LED because the LED is 3.5V and it is 150 ohm.

3.5.6 LED

Light emitting diode, used to detect if the power is on or off in the circuit.

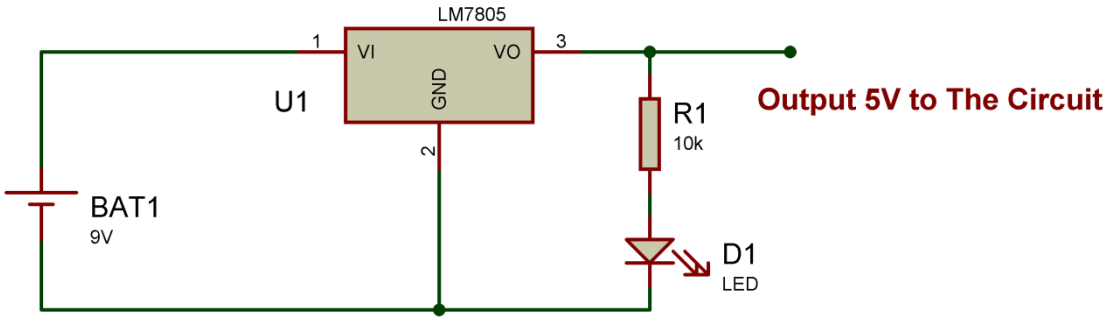


Figure3. 7 LM7805 Power Regulator Circuit

CHAPTER FOUR

SIMULATION AND RESULT

4.1Introuduction

In this chapter the simulation results will be illustrated along with the screen shots of the system, the system is simulated using Proteus simulation program.

4.2Simulation result

In order to verify the effectiveness of the system, a simulation model has been developed in Proteus plate form, under different operation condition.

4.2.1 Scenario one

The system first display the name of the thesis and please wait will be displayed till the system is initialized shows in figure (4.1)

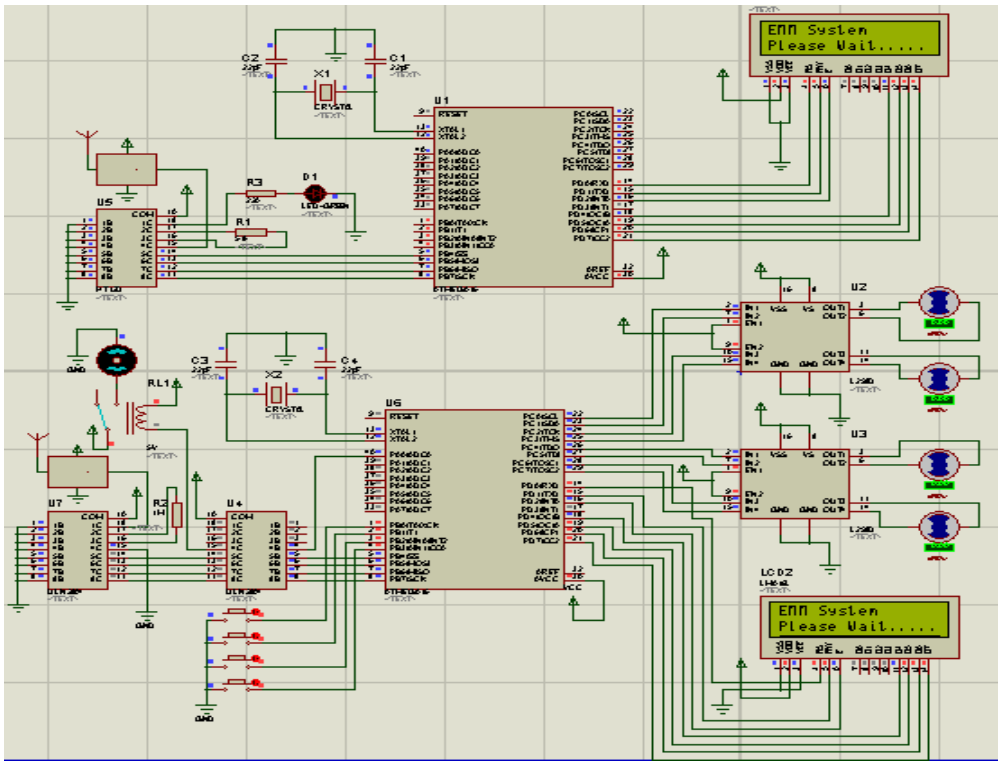


Figure4. 1 EMMS Under scenario one

4.2.2 Scenario tow

While running the system, the total number of pills is set to 10 or each container, and each pill has its container, the container has a 10 pills only, the container attached to a dc motor with gear box to control the angle while rotating the motor, and time of pills are configured into the microcontroller program to prevent users from re adjusting the time and interval of the pills.

The system display the number of the pills and the process that is happened while decreasing the number of pill from 10 to 9 pills and rotates the motor to get the pill show in figure (4.2).

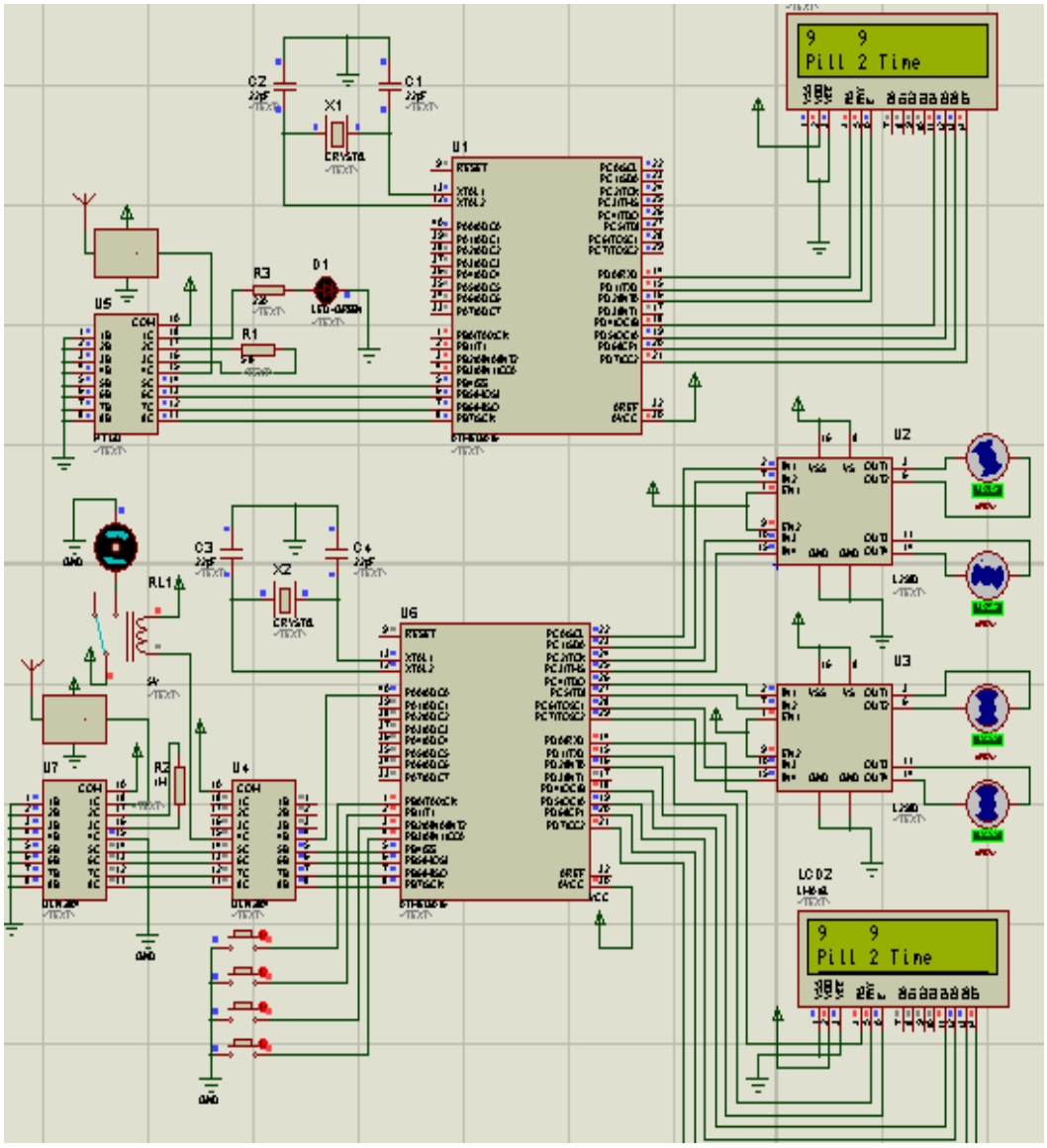


Figure4.2 EMMS Under scenario tow

4.2.3 Scenario three

A water container was also connected to solenoid valve controlled through microcontroller using an interface circuit consist of relay and current amplifier (ULN2804). The system displays the number of pills on the LCD, and wait for the result of the comparator that compare the reference time pills and the actual time in the microcontroller. Moreover the time of the microcontroller is set using timers with an accuracy of 1 millisecond (MS). Each time the reference equals the time of microcontroller the specific motor start rotating till a pill comes out. And then a solenoid valve is opened to fill water cup.

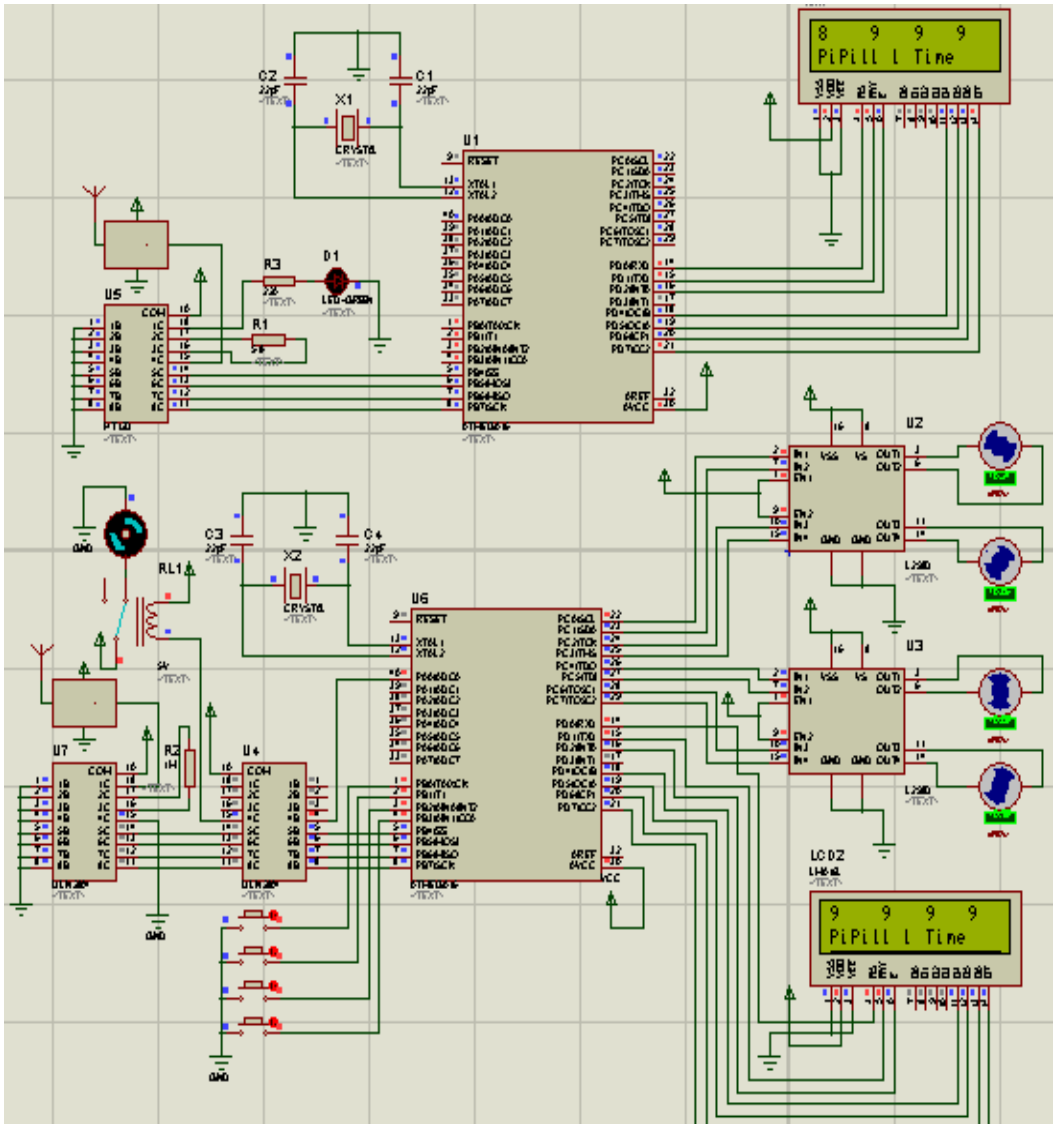


Figure4. 3 EMMS Under scenario three

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The simulation was successfully done and tested, with high accuracy of the system, except the transmitter and receiver because it does not have a simulation in the circuit simulation program, the alarm circuit can be a wireless system and the maximum distance is 80 meter with barriers and about 150 meter without barrier.

Limited number of components was interfaced to the microcontroller to reduce the design and optimize the circuit size.

The main problem faces the project is the feedback from the device to the nurse monitoring room to insure that the patient take the pills at its time from the device.

5.2 Recommendations

1. Add a feedback room device and patient to the monitoring server.
2. Use Bluetooth communication because it is non-interfering frequencies with medical devices.
3. Use servo motors for accurate angle and position.
4. Backup batteries should be added to the system in case of power failure.
5. Interface the microcontroller with network connector to use RJ45 to connect to any network.
6. Interfacing the circuit to pc to save records with time stamp.

Reference

[1] Evelyn Arnold, Guano Henry A, Feb 26, 1952, US Patent, US2587147 A

[2] Joseph Anthony Cappuccilli, 2 Aug 1977, US Patent, US4039080 A

[3] Richard Collens,2 Jun 1987, US Patent, US4669613 A

[4] Bristol-Myers Squibb Company, Dec 4, 1990, US Patent, US4974729 A

[5] Horowitz, Paul, Winfield Hill, "The Art of Electronics", Cambridge University Press, 1989.

[6] Texas Instruments, "LM2825 Integrated Power Supply 1A DC-DC Converter", 2010.

[7] Dorf, Richard C, Svoboda, James A, "Introduction to Electric Circuits", fifth edition, New York, John Wiley and Sons, 2001.

[8] E. Fred Schubert, "Light-Emitting Diodes", second edition, Cambridge University Press.

[9] Herman, Stephen, "Industrial Motor Control", sixth edition, Delmar, Cengage Learning, 2010.

[10] "Principle and application of Electrical Engineering", Giorgio Rizzoni, The Ohio state university,1995.

[11] Robert H Chen, "Liquid Cristal Display Fundamental Physic and Technology", Willy co, 2011.

[12] SGS-Thomson Companies, "Push-Pull Four channel Driver With Diodes", SGS-Thomson Microelectronics, Italy ,1996.

[13] A.P Godse, "Microprocessor and Microcontroller System", Technical Publication Pune,2007.

[14] Kenneth J Ayala, "The 8051 Microcontroller Architecture, Programing and Application", Western Caroline University,1991.

Appendices

Appendix A receiver circuit

```
Chip type           : ATmega16
Program type        : Application
Clock frequency     : 8.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
#include <mega16.h>
#include <delay.h>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>
// Declare your global variables here
int count1,count2,count3,count4,s1,s2,s3,s4;
```

```
char lcdbuffer[16];

void main(void)

{

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=0

PORTA=0x00;

DDRA=0x01;

// Port B initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In

// State7=0 State6=0 State5=0 State4=0 State3=P State2=P State1=P State0=P

PORTB=0x0F;

DDRB=0xF0;

// Port C initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

PORTC=0x00;
```

```
DDRC=0xFF;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
```

```
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// LCD module initialization
lcd_init(16);
while (1)
    {
    sar:
    count1=10;
    count2=10;
```

```
count3=10;
count4=10;
 lcd_gotoxy(0,0);
 lcd_putsf("EMM System      ");
 delay_ms(100);

 lcd_gotoxy(0,1);
 lcd_putsf("Please Wait     ");
 delay_ms(50);

 lcd_gotoxy(0,1);
 lcd_putsf("Please Wait.    ");
 delay_ms(50);
 lcd_gotoxy(0,1);
 lcd_putsf("Please Wait..   ");
 delay_ms(50);


 lcd_gotoxy(0,1);
 lcd_putsf("Please Wait...  ");
 delay_ms(50);
 lcd_gotoxy(0,1);
```

```
 lcd_putsf("Please Wait.... ");
 delay_ms(50);
 lcd_gotoxy(0,1);
 lcd_putsf("Please Wait.....");
 delay_ms(50);
 lcd_gotoxy(0,0);
 lcd_putsf("                ");
 delay_ms(50);
 lcd_gotoxy(0,1);
 lcd_putsf("                ");
 delay_ms(50);
  goto stage1;


  };


while(2)
{
stage1:
 lcd_gotoxy(2,1);
 lcd_putsf("Pill 1 Time     ");
 delay_ms(50);
 PORTC.0=0x01;
```

```
delay_ms(100);
PORTA.0=0x01;
delay_ms(150);
PORTC.0=0x00;
PORTA.0=0x00;
 count1=count1-1;
lcd_gotoxy(0,0);
ftoa(count1,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);
lcd_gotoxy(0,1);
lcd_putsf("Pill 2 Time     ");
delay_ms(50);
PORTC.3=0x01;
delay_ms(100);
PORTA.0=0x01;
delay_ms(150);
PORTC.3=0x00;
PORTA.0=0x00;
count2=count2-1;
lcd_gotoxy(5,0);
ftoa(count2,0,lcdbuffer);
```

```
lcd_puts(lcdbuffer);
delay_ms(500);
lcd_gotoxy(0,1);
lcd_putsf("Pill 3 Time     ");
delay_ms(50);
PORTC.5=0x01;
delay_ms(100);

PORTA.0=0x01;
delay_ms(150);
PORTC.5=0x00;
PORTA.0=0x00;
count3=count3-1;
lcd_gotoxy(9,0);
ftoa(count3,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);
lcd_gotoxy(0,1);
lcd_putsf("Pill 4 Time     ");
delay_ms(50);
PORTC.7=0x01;
delay_ms(100);
```

```
PORTA.0=0x01;
delay_ms(150);
PORTC.7=0x00;
PORTA.0=0x00;
count4=count4-1;
lcd_gotoxy(13,0);
ftoa(count4,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);
 if (count1<=2)
 {
  PORTB.4=0x01;
  delay_ms(100);
  PORTB.4=0x00;
  delay_ms(100);
 }
  if (count2<=2)
 {
  PORTB.5=0x01;
  delay_ms(100);
  PORTB.5=0x00;
  delay_ms(100);
```

```
        }
        if (count3<=2)
        {
        PORTB.6=0x01;
        delay_ms(100);
        PORTB.6=0x00;
        delay_ms(100);

        }
        if (count4<=2)
        {
        PORTB.7=0x01;
        delay_ms(100);
        PORTB.7=0x00;
        delay_ms(100);
        }
      };
}
```

Appendix A transmitter Circuit

Chip type : ATmega16

Program type : Application

Clock frequency : 8.000000 MHz

```
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>


// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>

// Declare your global variables here
int count1,count2,count3,count4,s1,s2,s3,s4;
char lcdbuffer[16];
```

```
void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=0
PORTA=0x00;
DDRA=0x01;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=P State2=P State1=P State0=P
PORTB=0x0F;
DDRB=0xF0;

// Port C initialization
```

```
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
```

```
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

while (1)
      {
      sar:
      count1=10;
      count2=10;
      count3=10;
      count4=10;

       lcd_gotoxy(0,0);
       lcd_putsf("EMM System      ");
       delay_ms(100);

       lcd_gotoxy(0,1);
       lcd_putsf("Please Wait     ");
```

```
delay_ms(50);

lcd_gotoxy(0,1);
lcd_putsf("Please Wait.    ");
delay_ms(50);

lcd_gotoxy(0,1);
lcd_putsf("Please Wait..   ");
delay_ms(50);

lcd_gotoxy(0,1);
lcd_putsf("Please Wait...  ");
delay_ms(50);

lcd_gotoxy(0,1);
lcd_putsf("Please Wait.... ");
delay_ms(50);
```

```
lcd_gotoxy(0,1);
lcd_putsf("Please Wait.....");
delay_ms(50);
 goto ss1;

 };

 while(11)
 {
 ss1:

s1=PINB.0;
s2=PINB.1;
s3=PINB.2;
s4=PINB.3;

 if (s1==0x00)
 {

lcd_gotoxy(0,0);
lcd_putsf("System Ready    ");
delay_ms(150);
```

```
lcd_gotoxy(0,1);
lcd_putsf("               ");
delay_ms(50);

lcd_gotoxy(0,0);
lcd_putsf("               ");
delay_ms(50);

 goto stage1;
 }


if (s2==0x00)
 {
 lcd_gotoxy(0,0);
lcd_putsf("               ");
delay_ms(50);


lcd_gotoxy(0,0);
ftoa(count1,0,lcdbuffer);
```

```
lcd_puts(lcdbuffer);
delay_ms(100);

lcd_gotoxy(5,0);
ftoa(count2,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(100);



lcd_gotoxy(10,0);
ftoa(count3,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(100);



lcd_gotoxy(14,0);
ftoa(count4,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(300);

  lcd_gotoxy(0,0);
lcd_putsf("System Ready    ");
```

```
        delay_ms(150);

        lcd_gotoxy(0,1);
        lcd_putsf("              ");
        delay_ms(50);

        lcd_gotoxy(0,0);
        lcd_putsf("              ");
        delay_ms(50);


         goto stage1;
        }


if (s3==0x00)
        {
         goto sar;
        }


      }
```

```
while(2)
{
stage1:

 lcd_gotoxy(2,1);
 lcd_putsf("Pill 1 Time     ");
 delay_ms(50);

 PORTC.0=0x01;
 delay_ms(100);

 PORTA.0=0x01;
 delay_ms(150);



 PORTC.0=0x00;
 PORTA.0=0x00;

 count1=count1-1;
```

```
lcd_gotoxy(0,0);
ftoa(count1,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);



lcd_gotoxy(0,1);
lcd_putsf("Pill 2 Time     ");
delay_ms(50);

PORTC.3=0x01;
delay_ms(100);

PORTA.0=0x01;
delay_ms(150);



PORTC.3=0x00;
PORTA.0=0x00;

count2=count2-1;
```

```
lcd_gotoxy(5,0);
ftoa(count2,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);

lcd_gotoxy(0,1);
lcd_putsf("Pill 3 Time     ");
delay_ms(50);

PORTC.5=0x01;
delay_ms(100);

PORTA.0=0x01;
delay_ms(150);

PORTC.5=0x00;
PORTA.0=0x00;


count3=count3-1;

lcd_gotoxy(9,0);
```

```
ftoa(count3,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);



lcd_gotoxy(0,1);
lcd_putsf("Pill 4 Time     ");
delay_ms(50);

PORTC.7=0x01;
delay_ms(100);

PORTA.0=0x01;
delay_ms(150);

PORTC.7=0x00;
PORTA.0=0x00;



count4=count4-1;

lcd_gotoxy(13,0);
```

```
ftoa(count4,0,lcdbuffer);
lcd_puts(lcdbuffer);
delay_ms(500);


 if (count1<=2)
 {
  PORTB.4=0x01;
  delay_ms(100);
  PORTB.4=0x00;
  delay_ms(100);

 }

  if (count2<=2)
 {
  PORTB.5=0x01;
  delay_ms(100);
  PORTB.5=0x00;
  delay_ms(100);

 }
```

```
        if (count3<=2)
       {
        PORTB.6=0x01;
        delay_ms(100);
        PORTB.6=0x00;
        delay_ms(100);

       }

        if (count4<=2)
       {
        PORTB.7=0x01;
        delay_ms(100);
        PORTB.7=0x00;
        delay_ms(100);

        }



      };
}
```