Digital Pill to Monitor Medication Adherence

S. S. Sabarinath [#1], C. S. Shyam Sathish [#2], R. S. Vidhyasagar [#3], S. Sathyamoorthy [*4]

[#]UG Student, [*]Assistant Professor

[#*]Department of Biomedical Engineering, Dhanalakshmi Srinivasan Engineering College,
Thuraiyur Main Road, Perambalur 621212, Tamil Nadu, India.

Abstract **- Digital pill is basically a multichannel sensor used for remote biomedical measurements using micro technology. This is used for the real-time measurement dosage level of tablet. The sensors are fabricated using electron beam and photolithographic pattern integration and were controlled by an application specific integrated circuit (ASIC). Digital pills are ingestible miniaturized electromechanical devices representing a point of convergence between biomedical technology, medicine and the pharmacy industry. Electronics, sensors and miniature robotic technology can give access, analyze and manipulate the body from the inside. The changes occurs in patient are monitored and sent it to nearby monitor for doctor monitoring through wireless.**

Keywords – Digital Pill, Application Specific Integrated Circuit, Photolithographic, Biomedical Technology, Miniature Robot Technology.

I. INTRODUCTION

A digital pill is a pharmaceutical dosage form that contains an ingestible sensor inside of a pill. The sensor begins transmitting medical data after it is consumed. The technology that makes up the pill, as well as the data transmitted by the pill's sensor, are considered to be part of digital medicine. The purpose of the sensor is to determine whether the person is taking their medication or not (called "adherence"). There are privacy concerns with respect to who receives the data and what is done with it.

It has not yet been established if these drug-device systems will positively contribute to medication adherence. Ability MyCite also comes with a disclaimer that data gathering should not be used for emergency situations, as tracking might be delayed or not happen at all. Experts on digital health, including Eric Topol of Scripps Translational Science, argue that it will take some time before this new health technology can impact adherence in a substantial way. People taking digital pills might, for example, refuse (or forget) to wear the accompanying patch that needs to be replaced every seven days to effectively collect signals from the swallowed pill.

AbilifyMyCite is a pill with an embedded sensor. Information (on ingestion, dose and batch of medication) is transmitted to a wearable skin patch (changed weekly) when the pill comes into contact with stomach acids. The information is then transmitted via Bluetooth to a smartphone application. Patients can share these data with family members and caregivers. The pill is metabolised and excreted like any other pill. The sensor was FDA-approved in 2012. Preliminary data on blood pressure and diabetes drugs, with this sensor as a stand-alone pill, show reported benefit to patients. AbilifyMyCite has not yet been proven to increase adherence and more studies are needed to confirm such correlation. It does not necessarily provide 'real-time' tracking because the detection of ingestion is said to take from half an hour to two hours, and transmission can be delayed or may not occur. Patients with abnormalities in stomach acidity might suffer inconsistencies.

Healthcare professionals must ensure that patients are willing and capable of using this system, and train them to use the drug, patch and phone application. Ethical issues have been raised: will such system be hacked and subsequently expose patients' private history of medications? Will children and those who are incompetent take those pills with no consent or must a judge authorise their use?

The approval of this digital pill may:

- Overcome the perennial challenge of patients adhering to medications;
- Open doors for other medication technology combos in the coming years;
- Facilitate the creation of libraries on adherence, which will significantly benefit patients and clinical trial data.

II. BLOCK DIAGRAM

A. *Transmitter*

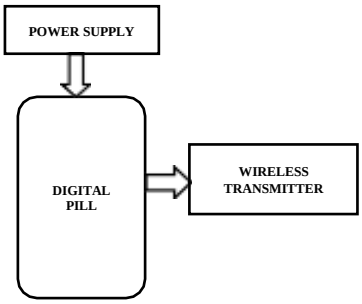

Fig. 1 Block Diagram of a Transmitter

B. *Receiver*

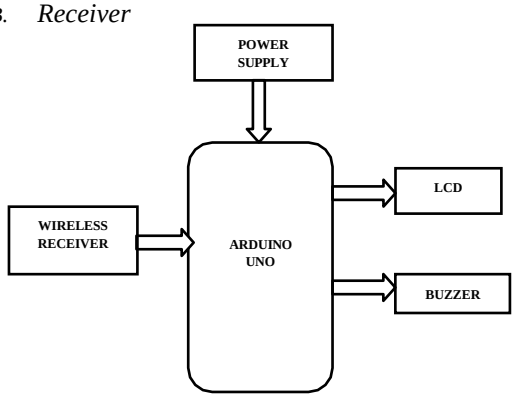

Fig. 2 Block Diagram of a Reciever

The digital pill which measures the dosage level of drug. The measured information is wirelessly transmitted to the receiver. The controller picks the information which is displayed on the LCD. When the high dosage level is received the buzzer goes on condition for indication purpose.

III. HARDWARE DESCRIPTION

A. *Arduino UNO*

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Specification

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volt
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

Communication

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required.

Fig. 3 Arduino Board

The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins.

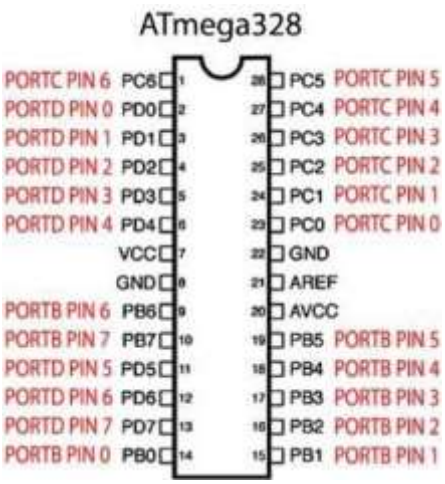

Fig. 3 PIN Configuration of ATmega328

B. ZIGBEE (Wireless Transmitter and Receiver)

It is a transreceiver module which provides easy to use RF communication at 2.4 GHz. It can be used to transmit and receive data at **9600 baud rates** from any standard CMOS/TTL source. This module is a direct line in replacement for your serial communication it requires no extra hardware and no extra coding. It works in Half Duplex mode i.e. it provides communication in both directions, but only one direction at same time. ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters line-of-sight, depending on power output and environmental characteristics. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee is typically used in low data rate applications that require long battery life and secure networking (ZigBee networks are secured by 128 bit symmetric encryption keys.) ZigBee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device. ZigBee was conceived in 1998, standardized in 2003, and revised in 2006. The name refers to the waggle dance of honey bees after their return to the beehive.

Zigbee devices are of three types:

1) ZigBee coordinator (ZC): The most capable device, the coordinator forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since it is the device that started the network originally. It stores information about the network, including acting as the Trust Center & repository for security keys.

2) ZigBee Router (ZR): As well as running an application function, a router can act as an intermediate router, passing on data from other devices.

3) ZigBee End Device (ZED): Contains just enough functionality to talk to the parent node (either the coordinator or a router); it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving long battery life. A ZED requires the least amount of memory, and therefore can be less expensive to manufacture than ZR or ZC.

Plus many other control and monitoring use as

ZigBee targets the application domain of low power, low duty cycle and low data rate requirement devices.

ZigBee is poised to become the global control/sensor network standard. It has been designed to provide the following features: – Low power consumption, simply implemented – Users expect batteries to last many months to years – Bluetooth has many different modes and states depending upon your latency and power requirements such as sniff, park, hold, active, etc.; ZigBee/IEEE 802.15.4 has active (transmit/receive) or sleep – Even mains powered equipment needs to be conscious of energy.

ZigBee devices will be more ecological than its predecessors saving megawatts at it full deployment. Low cost (device, installation, maintenance) Low cost to the users means low device cost, low installation cost and low maintenance. ZigBee devices allow batteries to last up to years using primary cells (low cost) without any chargers (low cost and easy installation). ZigBee's simplicity allows for inherent configuration and redundancy of network devices provides low maintenance. High density of nodes per network ZigBee's use of the IEEE 802.15.4 PHY and MAC allows networks to handle any number of devices. This attribute is critical for massive sensor arrays and control networks. Simple protocol, global implementation ZigBee's protocol code stack is estimated to be about 1/4th of Bluetooth's or 802.11's. Simplicity is essential to cost, interoperability, and maintenance. The IEEE 802.15.4 PHY adopted by ZigBee has been designed for the 868 MHz band in Europe, the 915 MHz band in N America, Australia, etc; and the 2.4 GHz band is now recognized to be a global band accepted in almost all countries.

Features

- Works on ISM band (2.4 GHz)
- No complex wireless connection software or intimate knowledge of RF is required to connect our serial devices.
- Designed to be as easy to use as wired ones.
- No external Antenna required.
- Plug and play device.
- Works on 5 DC supply.

Specifications

- Input Voltage - 5Volts DC
- Baud Rate - 9600
- RS 232 Interface & TTL Interface Range – Max 30 Mtrs - Line of Sight
- Channels - 3 Ch - JP1 & JP2 - Ch 1 On – On 2 CC2500 RF Module

Fig. 4 ZIGBEE Board

IV. SOFTWARE DESCRIPTION

A. *Arduino IDE*

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone.

Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.

The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project. The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

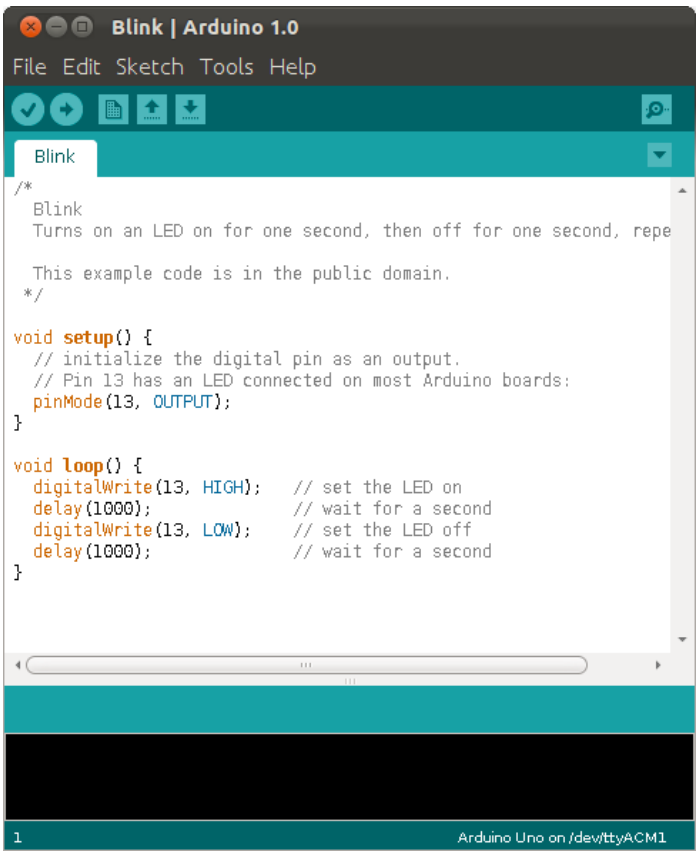

Fig. 5

Software Development

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board.

It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a

software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.

The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Software

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consists of only two functions:

Setup:

This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

Loop:

After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. Most Arduino boards contain a light-emitting diode (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.

B. PROTEUS

The microcontroller can understand a program written in assembly language, it must be compiled into a language of zeros and ones. Assembly language and Assembler do not have the same meaning. The first one refers to the set of rules used for writing program for the microcontroller, while the later refers to a program on a personal computer used to translate assembly language statements into the language of zeros and ones. A compiled program is also called Machine Code.

In machine code, the same command is represented by a 14-bit array of zeros and ones understandable by the microcontroller. All assembly language commands are similarly compiled into the corresponding array of zeros and ones. A data file used for storing compiled program is called an "executive file", i.e. "HEX data file". The name comes from the hexadecimal presentation of a data file and has a suffix of "hex" as well, for example "probe.hex".

After has been generated, the data file is loaded into the microcontroller using a programmer. Assembly language programs may be written in any program for text processing (editor) able to create ASCII data files on a hard disc or in a specialized work environment such as MPLAB described later.

Embedded C

Looking around, we find ourselves to be surrounded by various types of embedded system. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems.

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct execution of the program. Some ‘very fortunate’ developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.

Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications.

Embedded systems are programmed using different type of languages:

1. Machine Code
2. Low level language, i.e., assembly
3. High level language like C, C++, Java, Ada, etc.
4. Application level language like Visual Basic, scripts, Access, etc.

PROTEUS ISIS7 Suite

Proteus (**Pro**cessor for **Te**xt **E**asy to **Us**e) is a fully functional, procedural programming language which incorporates many functions derived from several other languages: C, BASIC, Assembly, Clipper/dBase; it is especially versatile in dealing with strings, having hundreds of dedicated functions; this makes it one of the richest languages for text manipulation.

Proteus owes its name to a Greek god of the sea (Proteus), who took care of Neptune's crowd and gave responses; he was renowned for being able to transform himself, assuming different shapes. Transforming data from one form to another is the main usage of this language.

Proteus was initially created as a multiplatform (DOS, Windows, UNIX) system utility, to manipulate text and binary files and to create CGI scripts. The language was later focused on Windows, by adding hundreds of specialized functions for: network and serial communication, database interrogation, system service creation, console applications, keyboard emulation, ISAPI scripting (for IIS). Most of these additional functions are only available in the Windows flavour of the interpreter, even though a Linux version is still available.

Proteus was designed to be practical (easy to use, efficient, complete), readable and consistent.

Its strongest points are:

- powerful string manipulation;
- comprehensibility of Proteus scripts;
- availability of advanced data structures: arrays, queues (single or double), stacks, bit maps, sets, AVL trees.

The language can be extended by adding user functions written in Proteus or DLLs created in C/C++.

IV. IMPLEMENTATION

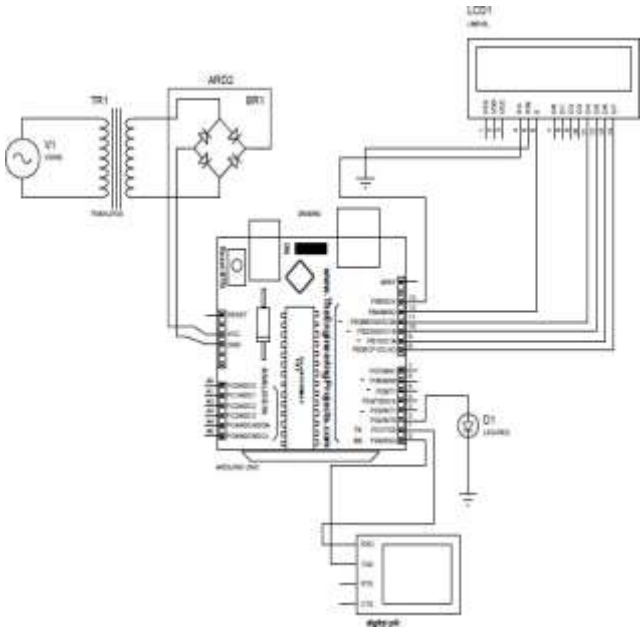

Fig. 6 Reciever Unit Circuit Diagram

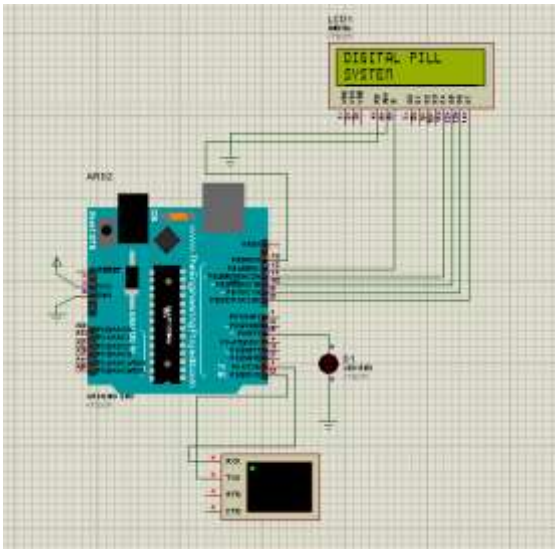

Fig. 7 Digital Pill System

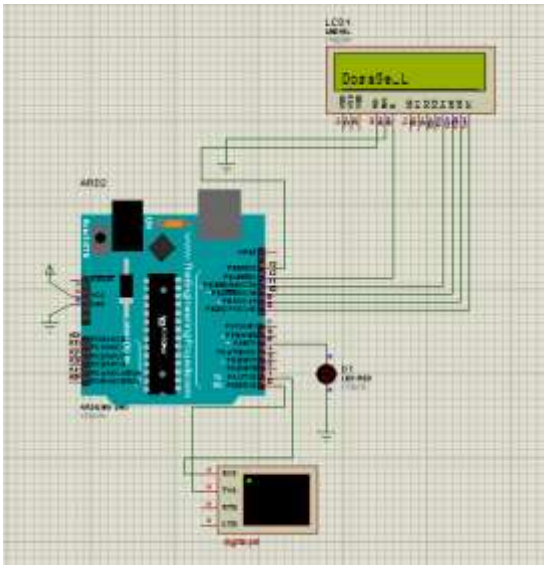

Fig. 8 Low Dosage

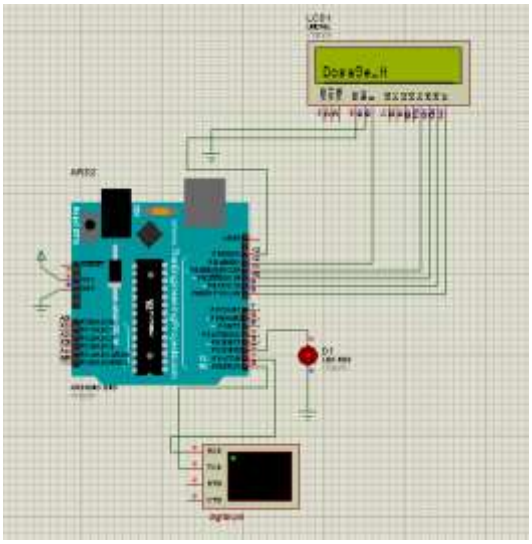

Fig. 9 High Dosage

V. RESULTS AND DISCUSSION

Observed Results

Stroke – unit care reduces mortality and improves functional independence. Stroke units might facilitate the maintenance of physiological homeostasis by bringing together medical, nursing and other staff experienced in stroke care; the necessary equipment for monitoring physiological parameters; and protocols to manage aberrant physiology. There is some evidence that intensive physiological monitoring in a stroke unit improves outcome. Improved outcome was also observed in a nonrandomized comparison between continuous

monitoring of ECG, temperature, oxygen saturation, limb movement and Heart rate.

Interpretation

The patient's data such as Accelerometer Co – ordinates, Body temperature, Heart beat rate, ECG value and SpO2 rate are obtained from the sensors such as Accelerometer, Temperature Sensor, Heart beat sensor, ECG Electrodes and SpO2 sensor respectively. The obtained values are received by the microcontroller, which calculates the percentage of occurrence of the disease consecutively. It then sends the message to the selected recipients and let them know the health condition of the patient in a real time basis, without the necessity of the recipients being either near the patient or near the kit.

Hence, the patient knows his/her health status from the LCD display, and the recipients could act accordingly to take any precautionary measures / treatment for the patient. The patient and the recipients also make sure that the status of occurrence of Ischaemic Cerebrovascular Stroke of the patient is continuously monitored.

VI. CONCLUSION AND FUTURE WORK

The dosage level already prefixed in the Application Specific Integrated Circuit (ASIC) can be matched with the dosage level of the medication under the experiment to find out the weather medication is taken or not or excessive is taken.The result that medication taken or not will be displayed in the LCD display .

Future Work

The development of digital capsule is possible by miniaturization of digital chips and other electronics devices. Especially CMOS and CCD technology should be further developed in future. The digital pill can also be designed with miniaturized cameras to track the path of digestive tract and the exact location of the pill
The main pressure is to reduce the components size which will release space that could be used for other capsule function. The new engineering methods for constructing tiny moving parts,miniature actuators and even motors into capsule endscope are being developed.

ACKNOWLEDGEMENT

This digital pill will prove useful for the patients those who are suffering from disorders like Bipolar Disorder, Schizophrenia, etc., These types of disorders are closely associated with loss of short term memory – patients those who consume medicines may forgot that they had consumed the appropriate levels of drug for the particular period, and consume the drug again, leading to overdosage and high concentration of Urine; the unwanted drug deposits in the kidney, making things worse. The digital pill monitors the levels of drug consumed and gives the feedback to such patients – they may record their activity by themselves.

For the Public, this device will also prove useful because nowadays, people seek attention over the quantity of supplementary drugs consumed, which affects the health and awareness quotient of an individual.

REFERENCES

[1] G. Ciuti, A. Menciassi, P. Dario, "Capsule Endoscopy: From Current Achievements to Open Challenges", IEEE Reviews in BiomedicalEngineering, vol. 4, pp.59-72, 2011

[2] G. Pan, L. Wang, "Swallowable Wireless Capsule Endoscopy: Progress and Technical Challenges", Gastroenterology Research and Practice, 2012

[3] E.A. Johannessen, Lei Wang, C. Wyse, D.R.S. Cumming, J.M. Cooper, "Biocompatibility of a Lab-on-a-Pill Sensor in Artificial Gastrointestinal Environments", IEEE Transactions on Biomedical Engineering, vol. 53,pp.2333-2340, 2006

[4] E. Stathopoulos, V. Schlageter V, B. Meyrat, Y. Ribaupierre, P. Kucera,"Magnetic Pill Tracking: A Novel Non-Invasive Tool for Investigation of Human Digestive Motility", Neurogastroenterology& Motility vol. 17, pp.148-154, 2005

[5] P. Hiroz, V. Schlageter, J.C. Givel, P. Kucera, "Colonic Movements in Healthy Subjects as Monitored by a Magnet Tracking System", Neuro- gastroenterology & Motility, vol. 21, pp.838-857, 2009

[6] J.L. Merino, C. Dehollain, "LC Tank Full Bridge Control for Large Coil Variations", 19th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp.653-656, 2012

[7] J.L. Merino, O.Kazanc, N. Brunner, V. Schlageter, M. Demierre, C. De- hollain, "Low Power Receiver for Magnetic Digestive Motility Tracking Pill", IEEE International Symposium on Circuits and Systems (ISCAS), pp. 469-472, May 2015

[8] J.L. Merino, C. Dehollain, "Predictive Calibration Technique for Magnetic Field Position Sensors", 5th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI), pp.136-141, 2013

[9] G. Pan and L.Wang, "Swallowable Wireless Capsule Endoscopy: Progress and Technical Challenges", Gastroenterology Research and Practice, vol. 2012, Article ID 841691, 9 pages, 2012.

[10] J.L. Merino, C. Dehollain, "LC Tank Full Bridge Control for Large Coil Variations", 19th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2012, pp 653-656, 2012.