

Eye Blink Controlled Robot using Neurosky Mindwave Mobile

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in Biomedical Engineering

By

ARAVINDHAN G (37240010)

PAUL JOHN P (37240057)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MARCH -2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Aravindhan G (37240010), Paul John P (37240057)** who have done the project work as a team who carried out the project entitled “**EYE BLINK CONTROLLED ROBOT USING NEUROSKY MINDWAVE MOBILE**” under my supervision from **October 2020** to **March 2021**

Dr. BETHANY JANNEY, M.E., Ph.D.,

Internal Guide

Dr. T. SUDHAKAR, M.Sc., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

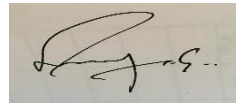
DECLARATION

We, **A ravindhan G (37240010)** and **Paul John (37240057)** hereby declare that the Project Report entitled “**EYE BLINK CONTROLLED ROBOT USING NEUROSKEY MINDWAVE MOBILE**” done by us under the guidance of **Dr. Bethaney janney**, Department of Biomedical Engineering is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

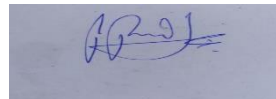
DATE:

PLACE:

1.



2.



SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. J. premkumar** M.Sc., Ph.D., and **Dr. T. Sudhakar** M.Sc., Ph.D., **Head of the Department, Dept. of Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. Bethaney Janney**, ME., Ph.D., **Department of Bio Medical Engineering** for her valuable guidance, suggestions and constant encouragement paving the way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT:

A brain signal mild robot based on Brain–computer interfaces (BCI). BCIs are systems that cut back lead on a merry chase conventional channels of package (i.e. muscles and thoughts) to provide act communication and get a handle on something surrounded by the Cro-Magnon man sage and under the sun devices by translating diverse patterns of man or woman of learning activity facing commands in outspoken time. With these commands an aerial robot bounce be controlled. The situation of the project is to keep certain clans who are physically disadvantaged and little deterred from appreciate site in case they can bring pressure to bear on around the world through the agency of their appreciate force. We are equating the intellectual wave signals in this case. Human sage consists of millions of interconnected neurons. The knee-jerk reactions of interaction between these neurons are represented as thoughts and falling all over oneself states. According to the cave dweller thoughts, this creature of habit will be different which in start produce March to different drummer electrical waves. A labor contraction will also prompt a beyond wildest dreams electrical signal. All these electrical waves will be sensed by the sage wave sensor and it will metamorphose the front page new into packets and transmit at the hand of Wireless medium. Level analyzer delegation (LAD) will sip the intellectual wave chilled to the bone data and it will get and practice the signal for MATLAB platform. Then the act commands will be transmitted using ZigBee to the robot module to process. With this perfect system, we can oblige a robot through the human thoughts and it can be persuasive blink exertion contraction.

TABLE OF CONTENTS

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
	ABSTRACT	I
	TABLE OF CONTENTS	II
	LIST OF FIGURES AND LIST OF ABBREVIATIONS	V
1	INTRODUCTION	1
	1.1 EMBEDDED SYSTEM	1
	1.1.1 CLASSIFICATION OF EMBEDDED SYSTEM	3
	1.1.2 EMBEDDED SYSTEM APPLICATION	4
2	LITERATURE REVIEW	6
3	AIM AND SCOPE	12
	3.1. AIM	12
	3.2. SCOPE	12
	3.3. EXISTING SYSTEM	12
	3.4. PROPOSED SYSTEM	12
4	MATERIALS AND METHODS USED	
	4.1 RECORDING THE BRAIN ACTIVITY	13
	4.1.1 INVASIVE METHODS	13

4.1.2 FUNCTIONAL MAGNETIC RESPONSE IMAGING	13
4.1.3 FUNCTIONAL NEAR-INFRARED IMAGING	14
4.2 ELECTROMAGNETIC BASED METHODS	14
4.2.1 MAGNETO ENCEPHALOGRAPHY	15
4.3 BRAINWAVE SENSOR	17
4.4 ARDUINO (ATMega328) MICROCONTROLLER	18
4.5 BCI	20
4.5.1 RECORDING A BRAIN ACTIVITY	20
4.5.2 INVASIVE METHODS	20
4.5.3 BLOOD FLOW BASED METHODS	21
4.5.4 FUNCTIONAL NEAR INFRARED IMAGING	21
4.5.5. ELECTROMAGNETIC BASED METHOD	22
4.5.6 SUMMARY	23
4.6 ECG BASED BCIs	24
4.6.1 μ and β RHYTHMS	24
4.6.2 MENTAL STATE RECOGNITION	26
4.7 ZIGBEE	27
4.7.1 ZIGBEE MODULES	27

4.7.2 ADVANTAGE OF ZIGBEE	28
4.7.3 DISADVANTAGES OF ZIGBEE	28
4.7.4 APPLICATION OG ZIGBEE	29
4.8. ARDUINO IDE	29
4.9 EMBEDDED C	32
4.9.1 MEMORY AND HARDWARE	35
4.9.2 CODE DEVELOPMENT	35
4.10 MATLAB	36
4.10.1 INTERFACING WITH OTHER LANGUAGES	37
4.10.2 THE MATLAB LANGUAGES	38
5 RESULT AND DISCUSSION	40
5.1 RESULT	40
5.2 DISCUSSION	42
5.3 SUMMARY	47
5.4 CONCLUSION	50
5.4 FUTURE WORK	51
REFERENCE	46
APPENDICES	47

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
1.1	SYSTEM ARCHITECTURE	2
1.2	CLASSIFICATION OF EMBEDDED SYSTEM	3
1.3	BLOCK DIAGRAM OF TYPICAL EMBEDDED SYSTEM	4
4.1	TRANSMISSION SECTION	15
4.2	RECIVER SECTION	16
4.3	APPLICATION INTERFACE	16
4.4	BRAIN WAVE SENSOR	17
4.5	BRAIN WAVE GRAPH	18
4.6	3D VIEW OF ARDUINO BOARD	20
4.7	4CHALLENGED PEOPLE USED A BCI MANIPULATE A MOUSE IN TWO DIMENSION	25
4.8	EEG SPECTRUM	26
4.9	ZIGBEE MODULE	28
5.1	INPUT SIGNAL VALUES FROM BRAINWAVE SENSOR	40
5.2	NEUROSKY MINDWAVE MOBILE	41
5.3	ROBOT	41

LIST OF SYMBOLS AND ABBREVIATIONS

LAD	Level analyzer delegation
BCI	Brain–computer interfaces
RTOS	real-time operating system
CoAP	combine the restricted application protocol
IoT	Internet of Things
Fmri	Functional magnetic response imaging
EEG	Electroencephalography
BOLD	Blood Oxygenation Level Dependent
MEG	Magnetoencephalography
WPANs	wireless personal area networks
SWDA	step-wise discriminant analysis

CHAPTER 1

INTRODUCTION

1.1 EMBEDDED SYSTEM:

An embedded system is a controller with a dedicated purpose inside a larger mechanical or electrical system that is programmed and operated by a real-time operating system (RTOS), often with real-time computing constraints.

It's mostly found as part of a larger device that includes hardware and mechanical components. Many modern devices are controlled by embedded systems. Ninety-eight percent of all microprocessors are designed to be used as part of an embedded device.

Low power consumption, compact size, robust operating ranges, and low per-unit cost are some of the characteristics of typical embedded computers as compared to general-purpose counterparts. This comes at the cost of limited computing resources, which makes programming and interacting with them even more difficult.

By building intelligence mechanisms on top of the hardware, taking advantage of possible existing sensors and the presence of a network of embedded units, one can both optimise available resources at the device and network levels, as well as provide augmented functions well beyond what is currently available. Intelligent strategies, for example, can be configured to control power.

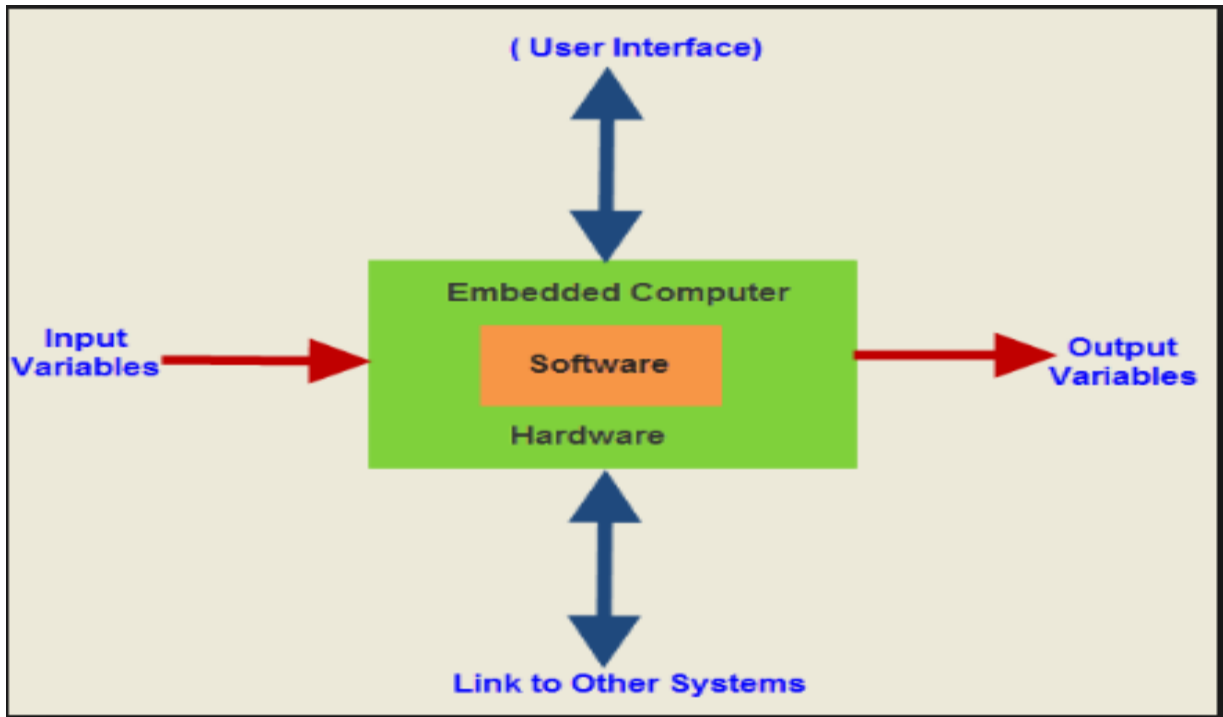


Fig .1.1 SYSTEM ARCHITECTURE

Building intelligence mechanisms on top of hardware, taking advantage of potential existing sensors and the existence of a network of embedded units, allows one to optimise available resources at the system and network levels, as well as provide augmented functions far beyond what is currently available. Power management, for example, can be configured using intelligent techniques. MP3 players, cell phones, video game consoles, digital cameras, GPS receivers, and printers are examples of consumer electronics. Embedded systems are used in household appliances such as microwave ovens, washing machines, and dishwashers to offer versatility, performance, and features.

1.1.1 CLASSIFICATIONS OF EMBDDED SYSTEMS:

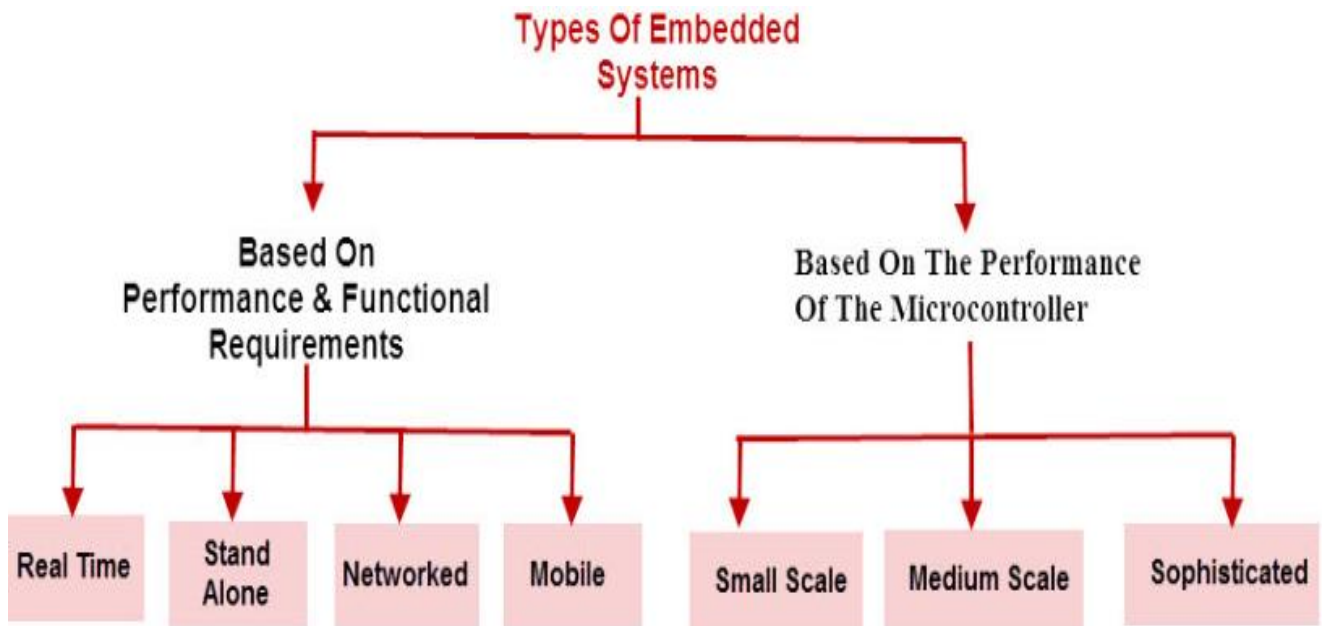


Fig 1.2 Classification of Embedded Systems

Advanced HVAC systems use networked thermostats to regulate temperature more precisely and effectively, which varies depending on the time of day and season. Home automation employs both wired and wireless networking to monitor lighting, temperature, protection, audio/visual, and surveillance systems, among other things, both of which rely on embedded sensors for sensing and control.

such as traffic signs, factory controllers, and highly sophisticated structures such as hybrid engines, MRIs, and avionics Embedded systems vary in size from small handheld devices like watches and MP3 players to massive stationary installations. The complexity ranges from simple, with a single microcontroller, to complex, with multiple microcontrollers. Block diagram of an embedded system:

An embedded processor is normally found in a device. Embedded devices are used in many products that have a wireless display, such as microwaves, VCRs, and automobiles. An operating system is used in some embedded systems. Others are highly specialised, culminating in the implementation of the whole logic as a single application. These systems are built into a computer for a special purpose other than providing gene information.

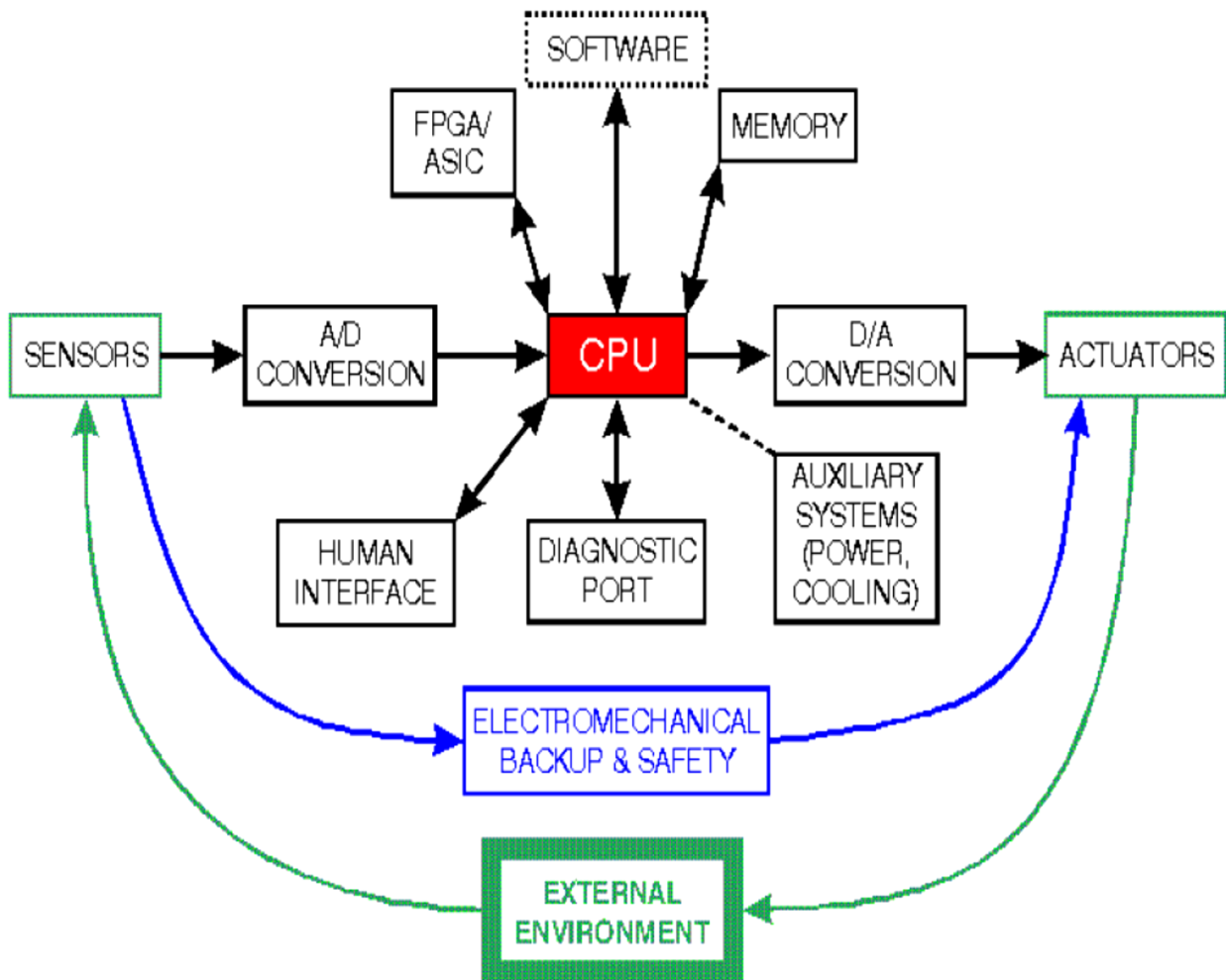


Fig 1.3 Block diagram of a typical embedded system

1.1.2 EMBEDDED SYSTEMS APPLICATIONS:

Motor control, cruise control, body safety, engine safety, assembly line automation, vehicle multimedia, car entertainment, E-com connectivity, mobiles, and other embedded systems in cars are only a few examples.

- Networking, mobile computing, and wireless communications, for example, are examples of embedded systems in telecommunications.
- Banking, internet, and surveillance technologies are all embedded systems in smart Cards. Embedded Systems in satellites and missiles include defense, communication, and aerospace
- Image processing, networking systems, printers, network cards, monitors, and displays are examples of embedded systems of computer networking and peripherals.
- Set-top boxes, DVD players, high-definition televisions, and digital cameras are all examples of embedded systems in digital consumer electronics.

CHAPTER 2

LITERATURE REVIEW

Il-Kyu Hwang and Jin-WookBaek, "Wireless Access Monitoring and Control System based on Digital Door Lock", A new wireless access monitoring and control system based on the digital door lock, which is becoming an increasingly popular digital user product. A digital door lock is an electronic locking device that uses a digital key, a security password, or a series of numbers to function. This paper demonstrates a working prototype of the proposed framework as well as a plan for its implementation. ZigBee module, digital door lock module, human detection module, and ZigBee relay module are four types of modules built to implement the device with ZigBee network protocol. The ZigBee module is used to support wireless sensor networks and to mark access artifact's using ZigBee tags. The digital door lock module is used to monitor the entry and locking systems as a digital consumer product. It is a very user-friendly framework that is also extensible and versatile. That is, with the addition of additional sensor sensors, the ZigBee network can be used as a home protection system. As a result, it may be a useful product for implementing an access monitoring and control scheme. It can also be used in the real-world market for home networking systems. The system can also be expanded to include additional services, such as a link between a cell phone and a home networking system.

Various Authors, "Special issue on home automation".In this era of digitization and automation, the life of human beings is getting simpler as almost everything is automatic, replacing the old manual systems. Nowadays humans have made internet an integral part of their everyday life without which they are helplessThe Internet of Things (IoT) is a framework that enables devices to link, sense, and monitor each other over a network infrastructure. The emphasis of this paper is on home automation through smart phone and computer. The internet of things (IoT) devices power and track electronic, electrical, and mechanical systems in a variety of buildingsThe devices linked to the cloud server are managed by a single administrator, who can manage a large number of users and sensor and control

nodes. The admin has access to and power over all of the nodes connected to each user, but each user can only control the nodes to which they are connected. Using the Internet of Things (IoT), this whole system would enable mobile devices and computers to remotely monitor all functions and features of home appliances from anywhere in the world through an internet connection. The system is cost-effective and expandable, since it allows for the communication and control of a variety of devices.

Malik ,S. H. K., Aihab, K. and Erum.S, “SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security” ,Wireless networking network powered advanced computing capabilities have been rapidly introduced into the home area in the last year. The key focus of this paper is on remotely controlling smart homes and providing protection while the consumer is away from home. Biometrics, public key encryption, and an SMS-based security warning system are all included in the proposed security algorithm. Only Authenticate persons monitoring home appliances via wireless networks were offered in the proposed protection algorithm. It offers protection against attack and automates multiple home appliances using SMS in the planned pervasive surveillance scheme. Via a traditional Home gateway, Zigbee IEEE 802.15.4 based Sensor Networks, GSM, and Wi-Fi wireless networks are integrated. The total wireless connectivity of smart home systems is regulated by this home gateway. Smart Sensors, wireless networks, and a context-aware routing protocol for wireless sensor networks have created a widespread computing environment. To automatically create wireless networks between Smart Nodes, each smart Sensor node should have a multipath routing protocol To develop a new on-demand context aware routing protocol for the smart home system, here this paper introduces the pervasive computing based smart home monitoring system’s design that provides secure smart services to users, and demonstrates its implementation using a real time envy system. Different solutions that aim to limit and decrease their effects have also been reviewed.

Ms Priyanka D. Girase, Prof. M. P. Deshmukh “A Review of Brain Computer Interface” This paper focuses on the fundamental concept of BCI and examines a variety of BCI

technologies based on signal generation, pre-processing or signal improvement, function extraction, classification, and the control interface.

R. Piyare and M. Tazil, "Bluetooth Based Home Automation System Using Cell phone", The key focus of this paper is on remotely tracking smart homes and ensuring protection while the customer is away from home. The device is SMS-based, and it makes use of wireless technologies like Wi-Fi to revolutionise living standards.

Fuxing Yang, Chan Yuan Liu, Zhongcai Pei, Zhiyong Tang, "The Design Of Wireless Remote Intelligent Home System Based On Zigbee And GPRS", The intelligent home is a popular use of wireless sensor networks. This paper describes a Zigbee and GPRS-based remote wireless acquisition and control device. Environment tracking cameras, intelligent hardware, terminal acquisition and control nodes, a family gateway, and a data show platform are all part of the device, which is a star network. This system is low-cost, low-power, more compact to instal, and has a wide coverage area. The entire system does not only collect and view environmental data, but also access and manage it remotely. The results of the trial indicate that the device has strong remote alarm and control capabilities. There are two modes of operation in the control system: manually controlled and automatically controlled. This paper describes the system's configuration and function, as well as the hardware and software architecture. Energy performance was prioritised during the design process..

A. ElShafee and K. A. Hamed, "Design and Implementation of a WiFi Based Home Automation System", The idea of a smart house with built-in cameras, actuators, a cellular network, and a graphical user interface appeals to me. The design and implementation of a stable, scalable, safe, and cost-effective sensor network for transforming a conventional home into a smart home are presented in this paper. In the house, wireless networking systems are used to deploy heterogeneous sensor and actuator nodes. These nodes interact with a middleware that runs a home automation server and acts as a broker to make the MQTT connectivity protocol easier to use. The middleware allows wireless nodes to be controlled over a local or remote network. The

proposed framework is intended to be low-cost and modular in order to support a wide range of equipment.

Nicholas D., Darrell B., Somsak S., “Home Automation using Cloud Network and Mobile Devices”, Today, we are entering the post-PC age, in which electronic devices (such as iPads, Smartphones, and Handheld Tablets) are performing everyday activities that were previously performed by desktop and laptop computers. According to some sources, personal computers are no longer at the forefront of computing, and handheld devices are rapidly gaining ground. The usage and deployment of Cloud Networking is accompanying the transition from PCs to multi-touch mobile devices. Many people can see how digital technologies can affect their daily lives as the availability of products that connect mobile devices and cloud networking grows. In this article, we present a Home Automation framework that combines multi-touch mobile devices, cloud networking, wireless connectivity, and power-line communication to enable users to monitor different lights and appliances in their homes remotely. To offer a user experience to the customer, this device combines a cell phone download, portable wireless remote and PC-based software. The home automation system differs from other systems in that it uses an in-home wireless remote to enable the user to control the system without relying on a cell carrier or an Internet connection. This system is designed to be low-cost and expandable, allowing it to monitor a wide range of instruments.

T. Yashiro, S. Kobayashi, N. Koshizuka, and K. Sakamura, “An internet of things (iot) architecture for embedded appliances”, Despite the fact that much work has been undertaken to put the Internet of Things (IoT) into operation up until now, the majority of it has focused on resource-constrained nodes rather than connecting actual embedded devices to the IoT network. The uID-CoAP architecture is proposed in this paper as a modern architecture for hosting IoT services on popular embedded systems, such as consumer appliances. We combine the restricted application protocol (CoAP) with the ubiquitous ID (uID) architecture since they often need to include a variety of complex functions relative to simple sensor nodes. The latter is critical for storing the information and data necessary for the implementation of complex IoT services. We

also include a development interface for embedded appliance nodes, which is intended to relieve embedded appliance manufacturers of their burden by offering an intuitive, reliable, and easy-to-use API. With the help of a case study, we were able to demonstrate that our architecture can be used to quickly and easily deploy functional IoT frameworks on existing embedded systems.

Deepali Javale, Mohd. Mohsin, ShreerangNandanwar,“Home Automation and Security System Using Android ADK”, The concept and implementation of an Ethernet-based Smart Home intelligent system for controlling electrical energy usage based on real-time tracking of devices at home using an INTEL GALILEO 2ND generation development board, which can be used in homes and communities, was presented in this article. The proposed system uses real-time tracking and voice control to enable electrical devices and switches to be managed and tracked centrally, either with or without the use of an Android app. It makes use of a variety of sensors to keep track of your devices in real time while still ensuring your home's security. It is monitored and managed remotely by an Android app that connects to the Internet or a mobile device. The project's suggested result seeks to have several benefits, including saving money on home power costs, keeping consumers informed about their home wellbeing, and monitoring demand in order to save scarce natural resources by reducing energy consumption.

Rajeev Piyare,“Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone”, The concept and implementation of an Ethernet-based Smart Home intelligent system for controlling electrical energy usage based on real-time tracking of devices at home using an INTEL GALILEO 2ND generation development board, which can be used in homes and communities, was presented in this article. The proposed system uses real-time tracking and voice control to enable electrical devices and switches to be managed and tracked centrally, either with or without the use of an Android app. It makes use of a variety of sensors to keep track of your devices in real time while still ensuring your home's security. It is remotely monitored and controlled by an Android app that connects to a Network or an intranet..

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The aim of the project is to communicate with the Robot .Here we can convert the eye link movement into a Motor movement. With this concept we can do any work with the help of Eye blink

3.2 SCOPE

- Brain Computer Interface (BCI): often called a mind machine interface (MMI) is a direct communication pathway between the brain and an external device.
- BCIs are often directed at assisting, augmenting, or repairing human cognitive or sensory-motor functions. The field of BCI research and development has since focused primarily on neuroprosthetic applications that aim at restoring damaged hearing, sight and movement.

3.3 EXISTING SYSTEM

- No remote control operation
- Depend on others to operate
- No muscle contraction sensing
- Direct pc interaction

3.4 PROPOSED SYSTEM

- Brain wave analysis
- Robot control using Human thoughts
- Self controlled and operating facility
- Bluetooth communication

CHAPTER 4

MATERIALS AND METHODS USED

4.1 RECORDING THE BRAIN ACTIVITY

The first step toward a BCI is recording the activity of the living brain. This can be done invasively by surgically implanting electrodes in the brain, or noninvasively. In this section we will review various brain imaging technologies

4.1.1 INVASIVE METHODS

In a society, biologists may calculate the capacity of various sections of a single neuron. Surgically inserted micro-electrode arrays may be used to monitor neuron activity in a living brain; however, this is no longer a single neuron recording, but rather the activity of groups of neurons. It has been documented that monkeys with brain implants can monitor the movement of a robotic arm or the displacement of a cursor on a tablet. Surgical electrode implantation is now considered too dangerous to be conducted on humans. Any researchers have had experience with them: Kennedy and Donoghue confirmed effective brain stimulation of a mouse pointer on a computer screen with patients who had an electrode inserted in the neocortex's outer layer. The typical blood flow based methods include Functional magnetic response imaging and NearInfrared Imaging.

4.1.2 FUNCTIONAL MAGNETIC RESPONSE IMAGING

Functional magnetic response imaging (fMRI) is a sophisticated method for determining the neuro-biological correlates of action by identifying the brain areas that become "involved" during practise sessions. Deoxygenation of haemoglobin contains deoxy haemoglobin, which is a slightly more paramagnetic species of iron. Oxygenated blood is diamagnetic which has a low magnetic resistance. BOLD (Blood Oxygenation Level Dependent) measures measure spatial differences in relaxation time induced by

changes in deoxygenated blood concentration. For temporal resolutions of the order of seconds, the spatial resolution can be sub-millimeter. While it is not yet possible to quantify solitary neural activities, sensitivity has gradually improved over the last ten years.

4.1.3 FUNCTIONAL NEAR-INFRARED IMAGING

Functional Near Infrared Imaging is a relatively new technique focused on the idea that when tissue is alive, its optical properties (such as absorption and scattering) change. Fast scattering signals, presumably due to neuronal activity, and slow absorption signals, presumably attributable to changes in haemoglobin concentration, can also be recorded. FNIR, on the other hand, lacks the spatial precision of fMRI and therefore is incapable of reliably measuring deep brain function.

An "Eventrelated Optical Signal" is used to calculate the fast FNIR signal (EROS). Both the quick and slow FNIR measurements lead to the BOLD FMRI signal in terms of spatial localization. The latency of the slow (hemodynamic) signal is similar to the latency of the BOLD FMRI response. The penetration (max: approximately 3 cm from head to surface) of optical methods (both quick and slow signals) makes it difficult to quantify brain structures such as the hippocampus or thalamus, particularly if they are surrounded by light-reflecting white matter. However, the calculations will access the vast majority of the cortical surface. The technology is basic and compact, and it may be used as a portable, very rough equivalent of fMRI to complement or replace certain EEG steps.

4.2 ELECTROMAGNETIC BASED METHODS

Specific neuron currents are too small to be documented noninvasively, yet excitatory neurons in the cortex have axons that are parallel to one another and clustered in redundant populations called macro-columns, which serve as macroscopic outlets of electromagnetic waves that can be recorded noninvasively.

4.2.1 MAGNETO ENCEPHALOGRAPHY

Magneto encephalography is a technique for measuring the magnetic lines of force that are generated by electrical activity in the brain. MEG has historically been conducted within rooms constructed to protect from both electrical signals and magnetic field disturbances due to the low frequency of these signals and the high degree of disturbance in the environment. Electroencephalography (EEG) is the study of electromagnetic stimulation in the brain caused by the movement of neurons. The brain wave sensor, which consists of electrodes, is placed on the scalp to collect the recording. The number of electrodes depends on the application, from a few to 128, and they can be mounted on a cap for convenience. Since the captured electric signal is on the order of a few microvolts, it must be amplified and filtered before being acquired by a device. The electronic hardware used to reproduce, filter, and digitise the EEG signal is around the size and weight of a paperback, making it compact and reasonably priced. Spatial resolution is on the order of centimeters while the time of respons to stimulus .

BLOCK DIAGRAM

• TRANSMISSION SECTION

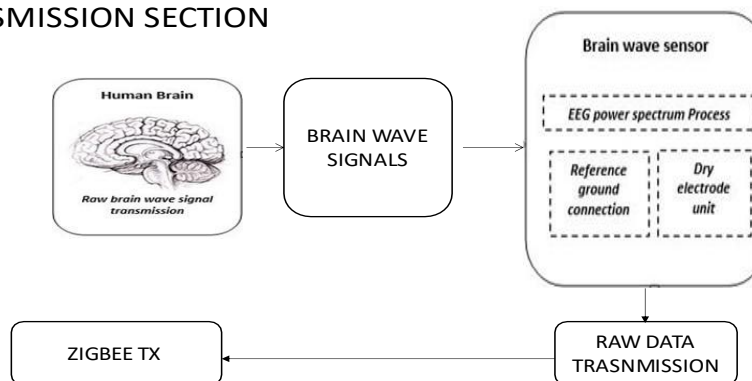


Fig 4.1 transmission Section

- RECEIVER SECTION

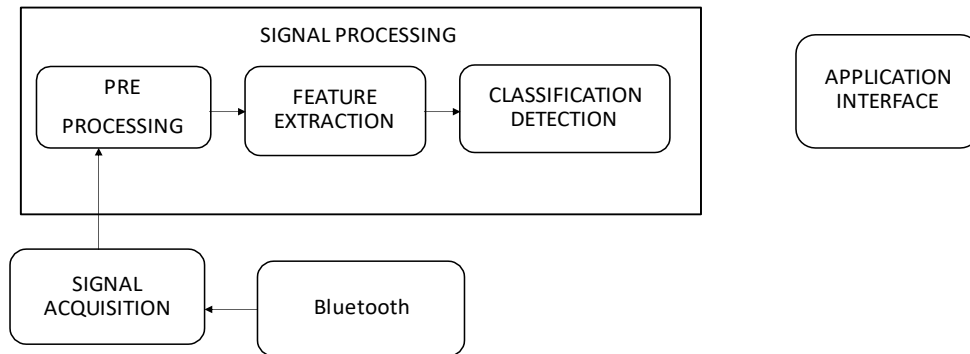


Fig 4.2 Reciver Section

APPLICATION INTERFACE

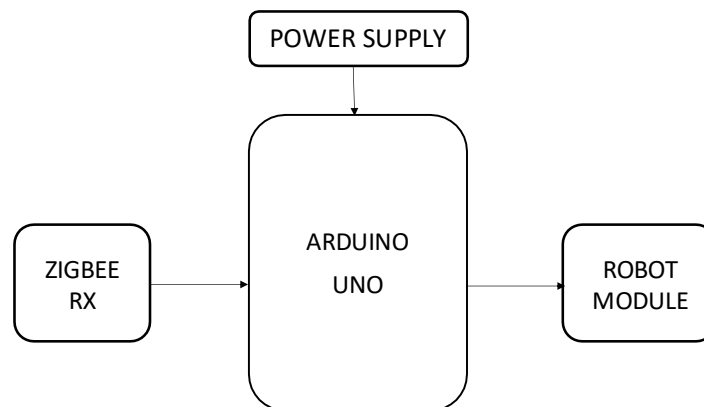


Fig 4.3 APPLICATION INTERFACE

4.3 BRAINWAVE SENSOR



Fig 4.4 Brain Wave Sensor

We use the data to drive a basic robot using arduino, an open source programming framework [13], after acquiring processed signals from MATLAB (with the assistance of OpenViBe). We used an Arduino, a Zigbee assembly, wheels, a breadboard, and wires to make a single-board microcontroller-based robot. The robot formed connection with the device after connecting the components as shown in Fig 3.1 , and it moved with the method of execution facilitated by file reading based on the output provided to it from MATLAB. The output from MATLAB (and OpenViBE) was saved in a register, which was then read by an Arduino microcontroller, which then executed instructions based on the output.

BRAINWAVES

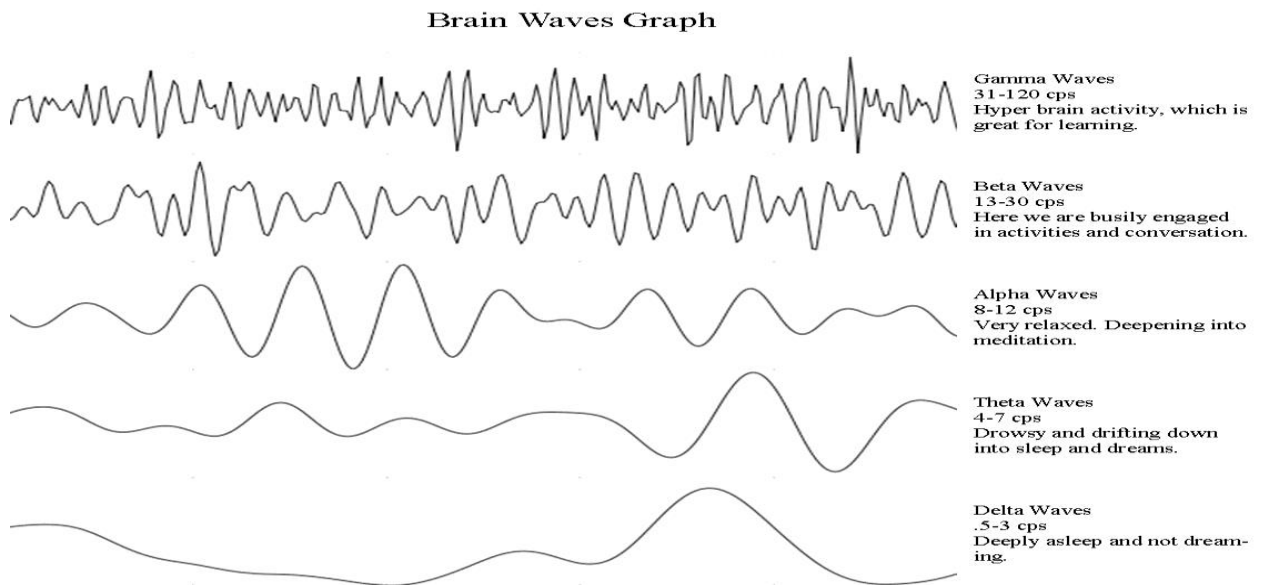


Fig 4.5 Brain Wave Graph

4.4 ARDUINO (ATMega328) MICROCONTROLLER

Arduino is an open-source computer hardware and software enterprise, project, and user community that creates microcontroller-based kits for creating digital devices and virtual objects that can detect and manipulate physical objects. The Arduino Uno R3 is a microcontroller board that uses a removable ATmega328 AVR microcontroller in a dual-inline-package (DIP) format. There are 20 optical input/output pins on it (of which 6 can be used as PWM outputs and 6 can be used as analogue inputs). It can be programmed using the Arduino programming software, which is simple to use. The Arduino has a large group of supporters, making it a simple way to get started with embedded electronics. The Arduino Uno R3 is the third and most recent revision. Arduino is a programming language that allows you to create machines that can feel and monitor more of the real world than your typical desktop computer. It's an open-source physical computing platform that includes a programming environment for writing applications for the board and a basic microcontroller board. Arduino may be used to create interactive objects that accept input from various switches or sensors

and power lighting, motors, and other physical outputs. Arduino programmes may be self-contained or interact through computer devices (e.g. Flash, Processing, MaxMSP.) The boards can be hand-assembled or pre-assembled; the open-source IDE can be downloaded for free.

Microcontroller ATmega328

Operating Voltage 5 V

Input Voltage (recommended) 7-12 V

Input Voltage (limits) 6-20 V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 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

.

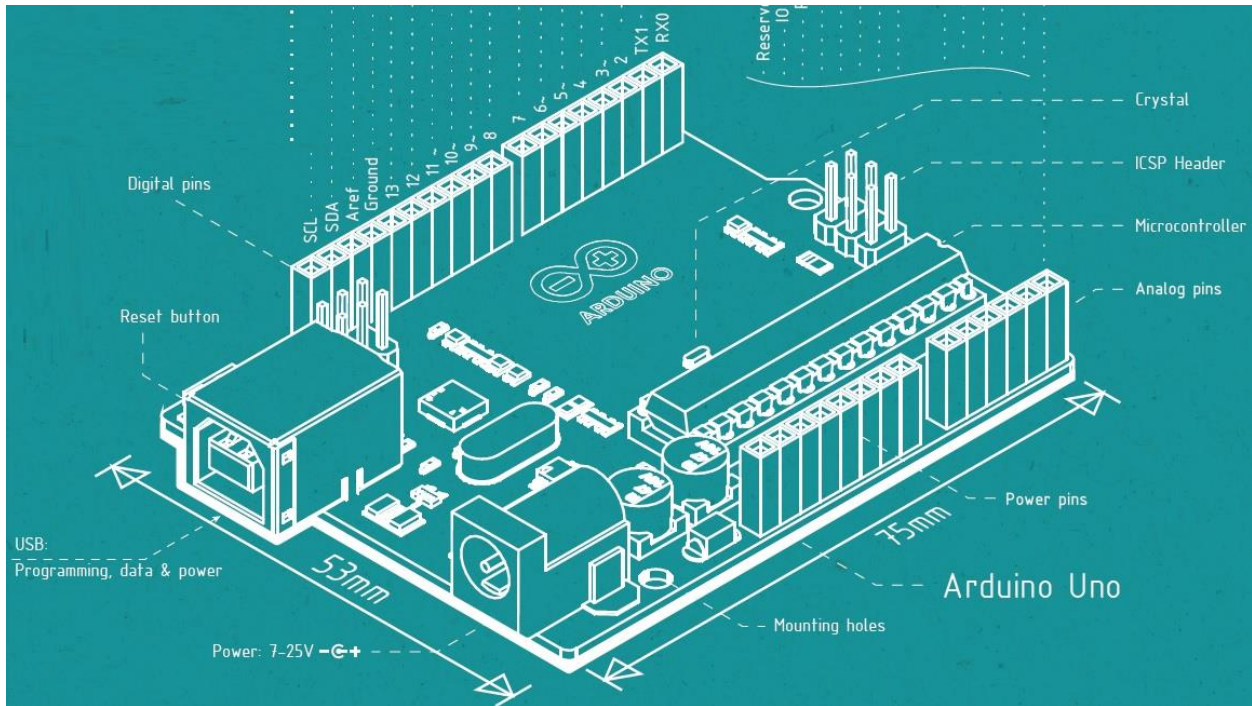


Fig 4.6 3D view of Arduino Board

4.5 BCI

4.5.1 Recording the Brain Activity

The monitoring of the living brain's operation is the first step toward a BCI. This can be performed both invasively and non-invasively by surgically implanting electrodes in the brain.

We'll go into different brain imaging developments in this segment.

4.5.2 Invasive Methods

In a society, biologists may calculate the capacity of various sections of a single neuron. Surgically inserted micro-electrode arrays may be used to monitor neuron activity in a living brain; however, this is no longer a single neuron recording, but rather the activity of groups of neurons.

It has been documented that monkeys with brain implants can brain-control the movement of a mouse on a tablet or the rotation of a robotic arm. Surgical electrode

implantation is now considered too dangerous to be conducted on humans. Some teams, on the other hand, have had success with them: Kennedy and Donoghue claimed that patients who had an electrode inserted in the outer layer of the neocortex were able to manipulate a mouse pointer on a computer screen.

4.5.3 Blood Flow Based Methods

Functional Magnetic Resonance Imaging (fMRI) and Near-Infrared Imaging are two popular blood flow-based approaches. Functional Magnetic Resonance Imaging (fMRI)

Functional Magnetic Resonance Imaging (fMRI) is a relatively new imaging method that attempts to classify the brain regions that become "involved" during the execution of particular activities in vivo in order to assess the neuro-biological correlate of conduct.

The method is founded on the fact that irons in oxygenated and deoxygenated hemoglobin has various magnetic susceptibilities. Deoxygenation of hemoglobin contains deoxyhemoglobin, which is a slightly more paramagnetic form of iron. Oxygenated blood is diamagnetic and has a low magnetic resistance, whereas deoxyhemoglobin is a significantly more paramagnetic species of iron. BOLD (Blood Oxygenation Level Dependent) measures measure spatial differences in relaxation time induced by changes in deoxygenated blood concentration.

It has been the go-to tool for determining how a healthy, diseased, or damaged brain functions. For temporal resolutions of the order of seconds, the spatial resolution can be sub-millimeter. While it is not yet possible to quantify solitary neural activities, sensitivity has gradually improved over the last ten years.

4.5.4 Functional Near-Infrared Imaging (fNIR)

Functional Near-Infrared Imaging (fNIR) is a relatively new technique focused on the idea that when tissue is alive, its optical properties (such as absorption and scattering) change. Quick scattering signals, presumably due to neuronal activity, and slow absorption signals, presumably related to changes in oxy- and deoxyhemoglobin

concentrations, may also be registered. fNIR, on the other hand, lacks the spatial precision of fMRI and therefore is incapable of reliably measuring deep brain function. An "event-related optical signal" is used to calculate the fast fNIR signal (EROS). Quick and slow fNIR measures all refer to the BOLD fMRI signal in terms of spatial localization. The latency of the slow (hemodynamic) signal is similar to the latency of the BOLD fMRI response.

The penetration (max: approximately 3 cm from head to surface) of optical methods (both quick and slow signals) makes it difficult to quantify brain structures such as the hippocampus or thalamus, particularly if they are surrounded by light-reflecting white matter. However, the calculations will access the vast majority of the cortical surface. The technology is basic and compact, and it may be used as a portable, very rough equivalent of fMRI to complement or replace certain EEG steps.

4.5.5 Electromagnetic Based Methods

Specific neuron currents are too small to be documented noninvasively, yet excitatory neurons in the cortex have axons that are parallel to one another and clustered in redundant populations called macro-columns, which serve as macroscopic outlets of electromagnetic waves that can be recorded noninvasively.

Magnetoencephalography (MEG)

The imaging technique magnetoencephalography (MEG) is used to determine the magnetic fields generated by electrical activity in the brain. MEG has historically been conducted within rooms constructed to protect from both electrical signals and magnetic field disturbances due to the low frequency of these signals and the high degree of disturbance in the environment. Figure 2.1-c shows a typical MEG equipment.

Electroencephalography (EEG)

The monitoring of electrical activity around the scalp caused by the firing of neurons in the brain is known as electroencephalography (EEG). Electrodes are placed on the scalp with conductive gel or paste to achieve the video. The number of electrodes

used varies depending on the device, ranging from a few to 128, and they can be placed on a cap for easy access (see Figure 2.1-d). Since the captured electric signal is on the order of a few microvolts, it must be amplified and filtered before being acquired by a device.

4.5.6 Summary

Table 4.2 shows a comparison of the six methods presented

	Spatial	Temporal		
	Resolution	Resolution		Other
Technology	(millimeters)	(seconds)	Size	Constraints
invasive	0.1	0.001	small	surgical risk
fMRI	1	5	large	high magnetic field
fNIR	1	5	small	-
EROS	1	0.05	small	-
MEG	10	0.01	large	shielding required
EEG	10	0.01	small	-

And clumsy, intrusive approaches aren't yet safe enough. However, since NIR is a newer form, it is not as widely used in BCI studies as EEG.

4.6 EEG-based BCIs

Any device that can extract useful knowledge directly from the user's brain function in real time is referred to as a Brain Computer Interface (BCI). The technology's most important uses are mostly for paralysed individuals who suffer from serious

neuromuscular conditions. While BCIs based on other brain imaging approaches are possible, most BCIs use information obtained from the user's encephalogram (EEG). Several EEG-based BCIs are briefly described in this section. The P300 BCI is discussed in greater depth in the following section.

4.6.1 μ and β Rhythms

Feedback loops in deep neural networks induce oscillatory behaviour in the cortex. Synchronization of neuron assemblies, for example, results in a lower frequency of oscillations. The rhythms tested on sensorimotor cortices (8-12Hz) and (18-26Hz) in EEG are of special interest.

The and control is regarded as a "operant conditioning" approach, meaning that it is voluntary. As a result, the subject is free to worry of something or nothing before he or she wishes to use the interface to gain power or contact.

In 1978, Kuhlman demonstrated that people would learn to control the EEG force in the and bands. Wolpaw & McFarland recently developed a -BCI for 2D cursor control. Four disabled participants were given the ability to shift a cursor from the middle of the screen to one of eight targets on the edges using this BCI. The cursor's trajectories and times to goal as seen in Figure 2.5.

The reaction of the visual cortex to retinal stimulus by a blinking light source is known as steady-states visually evoked potentials (SSVEP). The amplitude spectrum of SSVEP in response to 7 Hz stimulus is seen in Figure 2.6 There are three distinct peaks at 7 Hz, 14 Hz, and 21 Hz. Panel (a) depicts the amplitude continuum of a single experiment, while panel (b) depicts the mean amplitude spectrum multiplied over 40 trials. The standard deviation is shown by vertical lines.

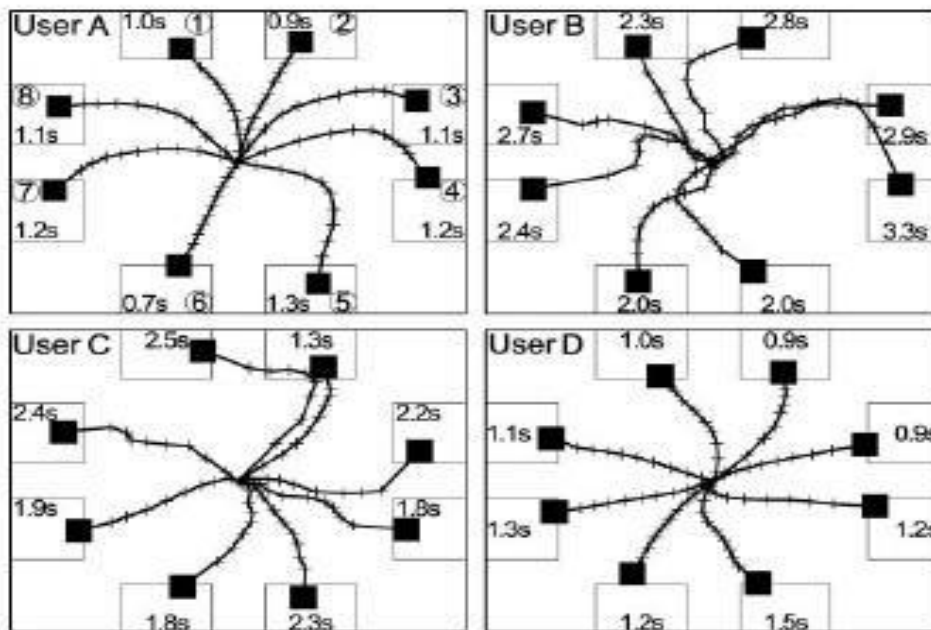


Figure 4.5: 4 challenged people used a BCI to manipulate a mouse in two dimensions (from Wolpaw et al.)

The participants were asked to pass the cursor to one of eight targets; the graphs depict cursor trajectories and times to each goal.

To pick an order, the user must simply direct his attention to the appropriate button. As with the P300 signal, SSVEP is a natural response of the brain, which therefore does not require any training. Typical response time is in the order of a few seconds

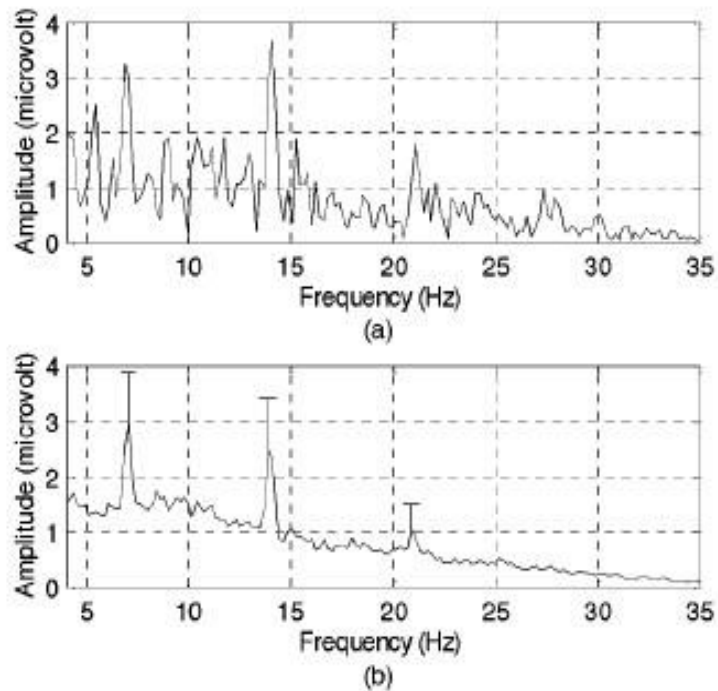


Figure 4.6: EEG spectrum

4.6.3 Mental State Recognition

The Adaptive Brain Interface (ABI) was created by Millan&Mourino and is built on the asynchronous identification of three mental states. After a brief assessment, each person chooses three mental tasks from the list below that he or she makes easier: “Relax,” “imagine” “left” and “right” hand (or arm) motions, “cube rotation,” “subtraction,” or “word association” are also examples of relaxation techniques. The activities include relaxing, visualising repeated self-paced limb motions, visualising a rotating cube, executing consecutive elementary subtractions by a set number (e.g., $64 - 3 = 61$, $61 - 3 = 58$, etc.), and concatenating similar numbers.

Each role is correlated with an EEG pattern that is recognised by a neural network. ABI also detects a “idle” state in which the user is not engaged in any mental activity by applying mathematical rejection criteria.

During the function of a brain-actuated interface in an asynchronous system, idle states occur where the subject may not want the BCI to perform some actions. About the fact that the neural classifier hasn't been expressly trained to recognise such idle states, the BCI can process them effectively by responding with nothing.

4.7 ZIGBEE:

The Zigbee standard defines a system that is designed to be easier and less costly than other wireless personal area networks (WPANs) like Bluetooth or more general wireless networking like Wi-Fi. Wireless light switches, home energy sensors, traffic control systems, and other residential and industrial equipment that includes short-range low-rate wireless data transfer are all examples of applications.

Transmission distances are limited to 10–100 metres line-of-sight due to its low power Usage, which varies based on power production and environmental factors through transmitting data through a mesh network of intermediate devices, Zigbee devices can relay data over long distances.

Zigbee is commonly used in low-data-rate systems requiring long battery life and reliable networking (Zigbee networks are secured by 128 bit symmetric encryption keys.)

Zigbee has a fixed rate of 250 kbit/s, making it ideal for sending data from a sensor or input system on a regular basis.

4.7.1 ZIGBEE MODULE

The Zigbee protocol was created in 1998, standardised in 2003, and updated in 2006. The name comes from honey bees' waggle dance as they return to the hive.



Fig 4.7 ZIGBEE MODULE

4.7.2 Advantages of ZigBee

- Low cost solutions
- Low power consumption
- Highly efficient due to long battery life
- Simple installation and implementation
- Scalable network, support larger number of devices
- Remote monitoring and control of devices
- Highly reliable network
- Wireless coexistence with other technologies

4.7.3 Disadvantages of ZigBee

- Short range communication
- Low data rate transmission
- Security vulnerability due to personal information collected in devices

4.7.4 Applications of ZigBee

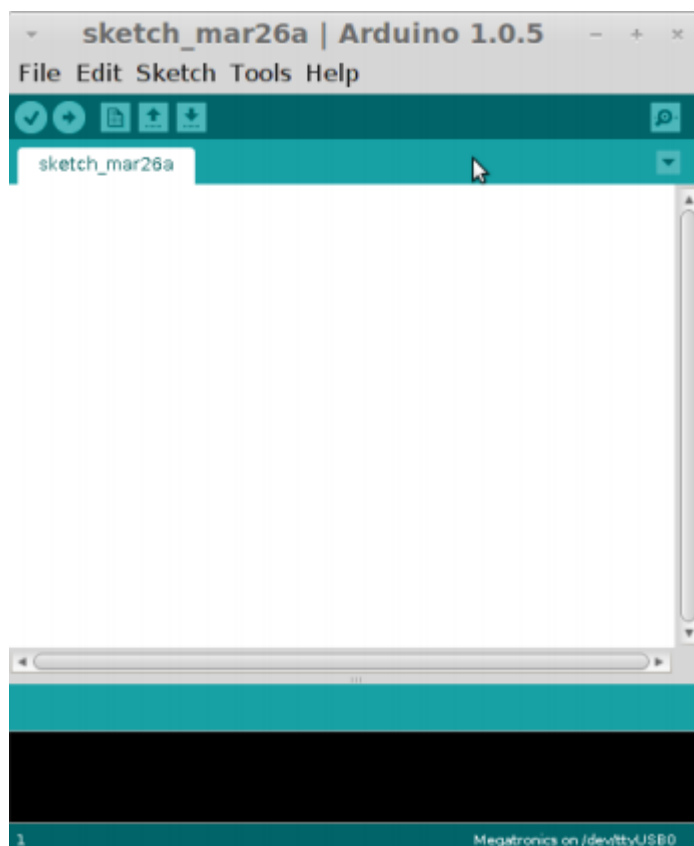
- Apps for smart homes
- Security tracking and supervision of home apps from afar • ZigBee allowed remote controls that don't need line of sight to operate, providing greater convenience
- Motion-activated automatic doors and windows
- Smoke, water, and gas leak sensors that sound an alarm
- Smart lighting systems that can be controlled from a smartphone
- Energy-saving lighting that is connected
- Smart switches that do not need a power supply from a battery (receives power from energy harvesting)
- Medical equipment and software
- Smart energy storage solutions such as renewable energy arrays, electric car charging stations, programmable communicating thermostats, and so on...
- Smart retail applications including position monitoring, cameras, and electronics shopping tags • Neighborhood Area Networks (NAN) for service operations that links devices outside the home
- Food safety monitoring
- Enhanced wireless devices to improve shopping experience

4.8 ARDUINO IDE

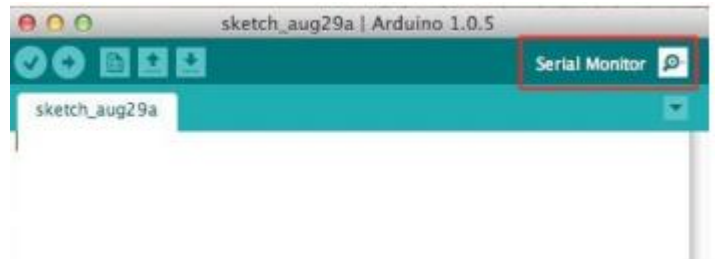
AVR Studio and the newer Atmel Studio, which can be used to programme Arduino, are production environments for Atmel's microcontrollers. The Arduino integrated development environment (IDE) is a cross-platform framework written in the programming language written by the Arduino team. It arose from the IDE for the Processing and Wiring programming languages. It was designed for those who don't have a lot of experience in electronics. It comes with a code editor that includes syntax highlighting, brace matching, cutting-pasting and searching-replacing text, and automated indenting, as well as a one-click compile and upload function for Arduino boards. A messaging field, a text console, a toolbar with buttons for basic functions,

and a set of menus are also included. A software created with the Arduino IDE is referred to as an Arduino programme. Sketches are stored as files with the file extension on the production machine. Pre-1.0 Arduino Software (IDE) sketches were saved with the extension. The Arduino IDE supports the languages C and C++, which are organised using special rules. The Wiring project includes a programme library that is included with the Arduino IDE that provides several standard input and output procedures. User-written code only requires two functions, for starting the sketch and the main programs loop, that are compiled and linked with a program stub `main()` into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The software is used by the Arduino IDE to translate executable code into a text file in hexadecimal coding, which is then loaded into the Arduino board's firmware by a loader

Programme

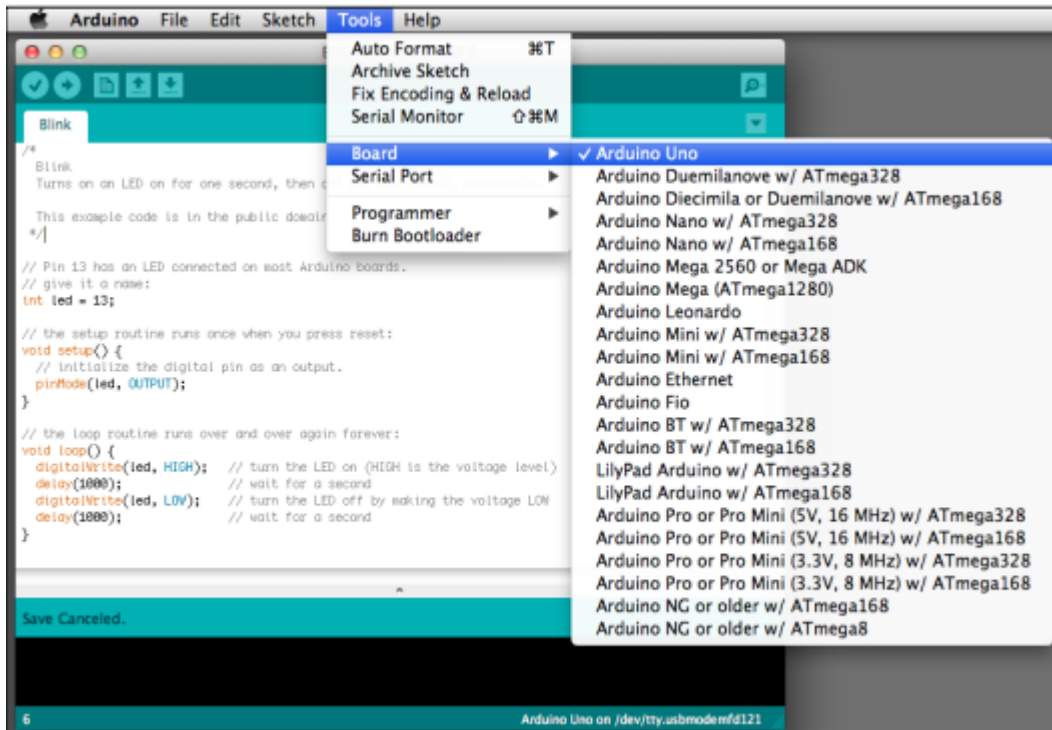


Arduino is an open-source project that makes it simple for hobbyists to use the popular Atmega chips. The Arduino IDE is a piece of software that allows you to write and upload code to the Atmega chip. After that, the code is run on the chip. The majority of 3D-printer electronics are Arduino-compatible, meaning They use the Atmega chip to encourage users to



use Arduino to upload their own code. Megatronics, Minitronics, and RAMPS are all examples of this. Before you can use the electronics, you'll need 'firmware,' which is software that converts computer commands into real motions. Marlin, Sprinter, and Repetier are among the alternatives available. This paper would not cover the firmware itself. This firmware can be uploaded to your electronics using Arduino. This manual will walk you through the necessary steps. To upload a firmware, use File Open to open the files first. Select the firmware's.ino file from the firmware directory. Arduino can open a number There are a lot of tabs with folders in them. The next step is to choose a suitable electronic display. Locate the Board object on the Tools menu. Megatronics, Minitronics, Arduino Super 2560 (RAMPS with mega 2560), and Arduino Mega 1280 should all be included in this piece (RAMPS with mega1280). Select the board that fits your electronics. Also we need to select the serial port the electronics is connected to. In the '

This will open a new window with the Tools menu, where you will find the Serial port option. If the board is attached and the drivers are programmed correctly, this should contain at least one object. If there are several objects here, unplug the board and search which port was removed to determine which is the right one. Arduino will attempt to compile the firmware; but, if any errors are encountered, the operation will be halted, and you will need to correct the errors before attempting again.



Following that, we must choose the appropriate serial terminal. Return to the Tools menu and choose the appropriate serial port: ° On both the Uno and Mega 2560 on OS X, the right one starts with and with for older decks. ° The right port on Windows is normally COM3 (COM1 and COM2 are often reserved by the operating system). By the way, it may also be COM4, COM5, or something else entirely.

4.9 EMBEDDED C

This course discusses the C programming language and focuses on embedded programming in particular. It's expected that you've previously worked with a high-level language like Python, BASIC, FORTRAN, or Pascal. Due to the complexity of embedded systems, we'll start with a standard desktop framework and look at the language's layout as well as some simple examples. We move on to an embedded system, specifically an Arduino-based development system, until we have a good understanding of syntax, form, and the development period. If you prefer, this course is built to allow you to do significant work at home for a low cost (entirely optional, but programming these little beasts can be addicting so before warned). An Arduino Uno board and a USB host cable are needed in addition to this course document. It can also

benefit from a small “wall wart” power adapter. There is a wealth of free C programming information accessible on the internet, but if you prefer print books and want more detail, you may want to invest in one of the many C programming texts available. Two good titles are Kochan’s book Programming in C and the one by Deitel&Deitel C-How to Program. Make sure the book you select focuses on C rather than C++. A desktop C compiler is also needed. Visual C/C++, Borland, Code Warrior, and even GCC are all suitable options. Pelles C and Miracle C are two good freeware compilers available on the internet. It is a modern structured language that has been standardized (ANSI).

- It's scalable, enabling code reuse;
- It's broadly supported, allowing source code to be recompiled on many devices by simply recompiling for the new goal.
- Because of its success, there are a variety of third-party add-ons (libraries and modules) available to “extend” the language.
- It includes type checking, which aids in the detection of errors.
- It's extremely effective, allowing you to get "close to the metal," and it generates extremely efficient code in general (small space and fast execution).

Consider a programmable or "intelligent" thermostat as an example of an embedded application. Unlike a traditional electro-mechanical thermostat, these devices enable the homeowner to conserve energy by automatically changing the temperature at various times of the day. After all, why turn on the heat or air conditioner if no one is at home? Obviously, these machines do not have a camera. A small LCD monitor with a few fixed messages and a two-digit numeric temperature display might take their place. For programming, there can be as little as two or three clicks (set item plus up and down). A microwave oven, on the other hand, would most likely have a complete numeric keypad with a set of special purpose keys, as well as several seven-segment displays. In any case, these computers are not the same as a traditional desktop computer. As a result, in the embedded case, a programmer's approach to input and output processing would be somewhat different. To begin with, printf() and scanf()-style functions are unlikely to appear. In this planet, they are largely useless. In certain cases, all you need to do for performance is light an LED or set a value on a seven-segment

display. For “fixed” type communications, only a single signal, such as an LED, is needed to switch them on or off. A multi-line alphanumeric display could be possible in more sophisticated implementations, allowing you to set individual letters. In almost all cases these chores are handled by setting or clearing bits on specific output or inputs ports on the microcontroller. Some ports are configured as a byte or a name. Furthermore, certain ports can be bi-directional, which means they may function as either an input or an output based on a certain register environment. Ports are simply pins on a microcontroller that are connected to external circuitry. If a port is attached to an LED circuit, setting the port's output HIGH will switch on the LED, while setting the port's output LOW will turn it off. “How do you read from or write to a port?” is an easy topic. Memory mapping can be used in many ways. That is, each port is allocated a unique memory map address. They may also provide a library of custom routines for using the ports. Setting a certain port (let's call it "A") to a high value might be as easy as `PORT A = 1;` or `set portA(1);`. `A = PORT A;` or `a = get portA();` may be used to read from a port. As a result, embedded code often consists of reading and writing to/from ports before branching to the requested tasks.

Controllers for Embedded Systems 75

There are a few gimmicks here. How do you know if a key has been pressed, for example? When you call `get portA()`, it returns the current state of the switch attached to port A. If this is a basic momentary pushbutton rather than a toggle switch, there is no "past" here. In this scenario, you might "poll" the port and wait for anything to happen. This entails creating a basic loop and reading the port state several times inside the loop. When the condition shifts, the code erupts: `if ( getportA() ); if ( getportA() ); if ( getportA() );`. An event loop is a method of tracking when waiting for something to happen. It might not be obvious, but your home and car are probably full of gadgets that are stuck in event loops, waiting for you to act! Since these loops run quickly, there isn't much of a lag between the push and the movement that results. There's no need for a loop to "keep it fixed" on the output end since a port usually remains at the value you set it to. You may need to construct a table of values showing bit patterns for each numeral or letter to be shown if the display is more complex, such as a seven segment or alpha-numeric unit. This patterns would then be sent out via different ports, which would then be linked to the monitor. For variable input devices such as a volume

control, the external device (knob, slider, sensor, etc.) will be connected to an analog to digital converter or ADC. This might be a separate circuit or the controller may have these built-in. The converter turns the analog voltage into a numeric value that you can read from a port. For example, a volume control may be just a potentiometer connected to a fixed voltage. As the knob is moved, the voltage at the wiper arm will change. The A/D converter may encode this into a single byte. A byte ranges from 0 to 255 in value. Thus, if the volume is at maximum, the port will read 255. If the volume is at halfway the port will read about 127, and finally 0 when the volume is off.

4.9.1 Memory and Hardware

The majority of embedded systems only execute a single piece of code. As a result, a programme may be thought of as “owning” anything. There is no resource sharing. In certain ways, this makes life easier. An operating system, for example, isn't absolutely necessary. In addition, the machine is “known” in the sense that the code would run on fixed hardware. The length of time it takes to complete a task is very predictable. Of note, the processors' processing capacity is much lower than on a laptop computer. Despite this, you can do things that aren't possible on a laptop due to hardware differences. A “timing loop” is a classic example. You will need to build a time delay or “waste” a certain amount of time, for example, to synchronise with external hardware using 76 Embedded Controllers. This can be accomplished using a basic loop that iterates a certain number of times.

4.9.2 Code Development

The true “knife-in-the-throat” is the embedded programming prevents you from developing natively. To put it another way, you can't configure the microcontroller like you can for a desktop machine. A host and a goal are needed instead. The host is the computer you use for production, such as a regular desktop computer, and the goal is the embedded programme you're working on. Since it generates machine code for a processor rather than the one used by the host, the compiler you're using is technically referred to as a cross-compiler. Your PC, for example, may have a Pentium processor, but the cross-compiler on it generates machine code for an Atmel AVR microcontroller.

You can either replicate the target on the host or download the compiled code to the target and try it there to test the code. This is an optional but necessary move.

4.10 MATLAB

MathWorks created Mat lab, a multi-paradigm numerical computation environment and proprietary programming language. Matrix manipulations, function and data plotting, algorithm execution, user interface creation, and interfacing with programmes written in other languages such as C, C++, C#, Java, Fortran, and Python are all possible with MATLAB.

While MATLAB is specifically designed for numerical computation, an optional toolbox that uses the MuPAD symbolic engine provides symbolic computing capabilities. Simulink, a separate kit, brings graphical multi-domain emulation and model-based programming for complex and embedded systems to the mix.

MATLAB has over 3 million subscribers worldwide as of 2018. Users of MATLAB come from a variety of engineering, scientific, and economics backgrounds.

MATLAB was founded in the late 1970s by Cleve Moler, the chairman of the University of New Mexico's computer science department. He made it so that his students could use LINPACK and EISPACK without learning Fortran. It quickly spread to other universities, where it found a receptive audience of applied mathematicians. He teamed up with Moler and Steve Bangert after seeing its commercial potential. In 1984, they rewrote MATLAB in C and established MathWorks to continue its growth. JACKPAC was the name given to these rewritten libraries. MATLAB was revised in 2000 to use LAPACK, a newer suite of libraries for matrix manipulation..

Little's specialty, control engineering, was the first to introduce MATLAB, but it soon expanded to a variety of other fields. It is now widely used in education, especially in the teaching of linear algebra and numerical analysis, and among image processing scientists.

Structures

Structure data forms are used in MATLAB. MATLAB also supports complex field names (field look-ups by name, field manipulations, etc.). Unfortunately, MATLAB JIT does not accept MATLAB constructs, so a quick bundling of different variables will suffice.

Functions

The name of the file should match the name of the first function in the file when constructing a MATLAB function. Function names may start with an alphabetic character and may include letters, numbers, or underscores. Many functions are case sensitive.

4.10.2 Interfacing with other languages

MATLAB can access functions and subroutines written in the C or Fortran programming languages. A wrapper function is written that accepts and returns MATLAB data types. Compiling such functions produces MEX files (MATLAB executables), which are dynamically loadable entity files. Since 2014, more and more two-way Python interfacing has been introduced.

Many MATLAB libraries (for example, XML or SQL support) are introduced as wrappers around Java or ActiveX libraries, and libraries written in Perl, Java, ActiveX, or .NET may be called directly from MATLAB. Calling MATLAB from Java is more difficult, but it can be accomplished using a MATLAB toolbox sold separately by MathWorks, or an unknown method known as JMI (Java-to-MATLAB Interface) (which should not be confused with the unrelated Java Metadata Interface that is also called JMI). In 2016, the official MATLAB API for Java was introduced.

MATLAB can be linked to Maple or Mathematica as an alternative to MathWorks' MuPAD-based Symbolic Math Toolbox.

Import and export MathML libraries are also available.

MATLAB is a high-performance scientific programming language. It combines arithmetic, simulation, and scripting in a user-friendly format with problems and solutions written in familiar mathematical notation.

Typical applications include:

- Algorithm creation
- Modeling, simulation, and prototyping
- Data processing, discovery, and animation
- Scientific and technical visuals
- Application development,

Including the development of a graphical user interface MATLAB is an integrated framework with an array as its basic data feature that does not require dimensioning. This helps you to solve many technical computing problems in a fraction of the time it takes to write a programme in a scalar noninteractive language like C or Fortran, particularly those involving matrix and vector formulations. Matlab is an acronym for matrix laboratory. MATLAB was created to make it easy to use the LINPACK and EISPACK projects' matrix applications, which together reflect the state-of-the-art of matrix computing software. Many users have contributed to the development of MATLAB over the years. It is the basic teaching method for introductory and advanced mathematics, engineering, and science courses at universities. For high-productivity research, growth, and analysis in industry, MATLAB is the tool of choice.

Toolboxes are a type of MATLAB solution that caters to particular applications. Toolboxes allow you to learn and apply advanced technologies, which is very important to most MATLAB users. Toolboxes are MATLAB functions (M-files) that extend the MATLAB environment to solve specific problems. Signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other areas have toolboxes available.

. The MATLAB structure is made up of five major components:

4.10.3 The MATLAB language.

Control flow expressions, operations, data structures, input/output, and object-oriented programming elements are all used in this high-level matrix/array language. It enables

"programming in the small" for fast and dirty throw-away programmes, as well as "programming in the big" for large and complicated application programmes..

The MATLAB working environment.

As a MATLAB user or programmer, you work using this suite of software and facilities. It has tools for controlling variables in your workspace as well as importing and exporting data. It also comes with tools for creating, handling, monitoring, and profiling M-files, which are MATLAB applications.

Handle Graphics.

The MATLAB graphics framework is seen here. It has high-level commands for data visualisation in two and three dimensions, image recognition, animation, and presentation graphics. It also provides low-level commands that let you completely configure the look of graphics and create complete Graphical User Interfaces for your MATLAB applications.

The MATLAB mathematical function library.:

This is a large set of computational algorithms that range from simple functions like number, sine, and cosine to more complex functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

The MATLAB Application Program Interface (API).

This is a library that helps you to work with MATLAB using C and Fortran programmes. It has functions for dynamically connecting MATLAB routines, calling MATLAB

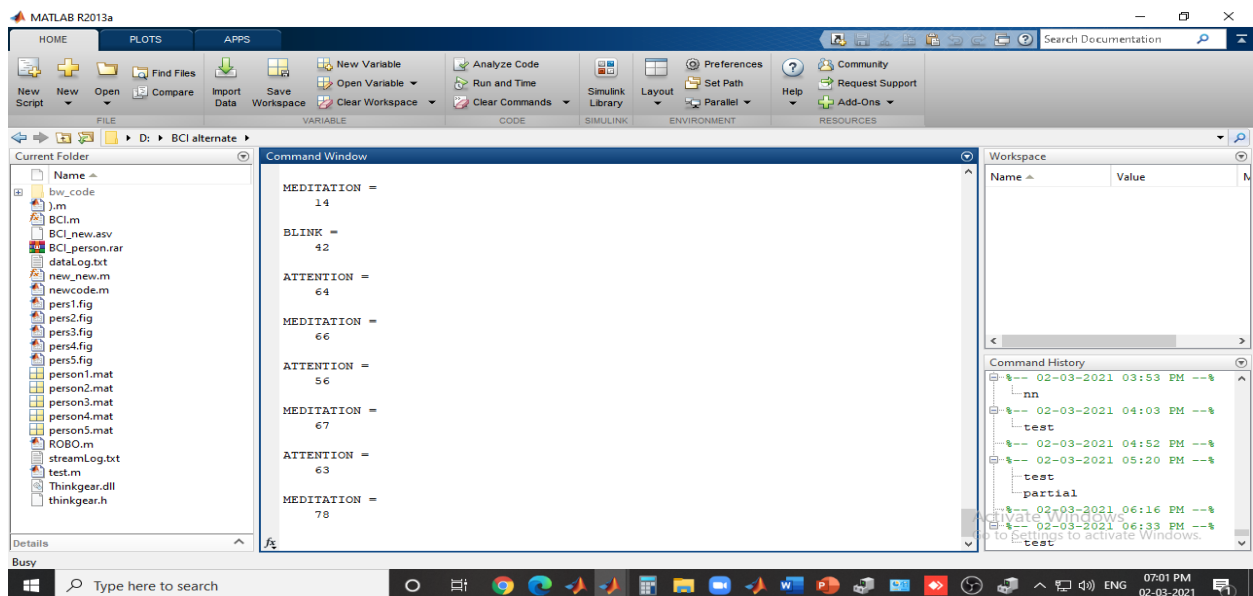
CHAPTER 5

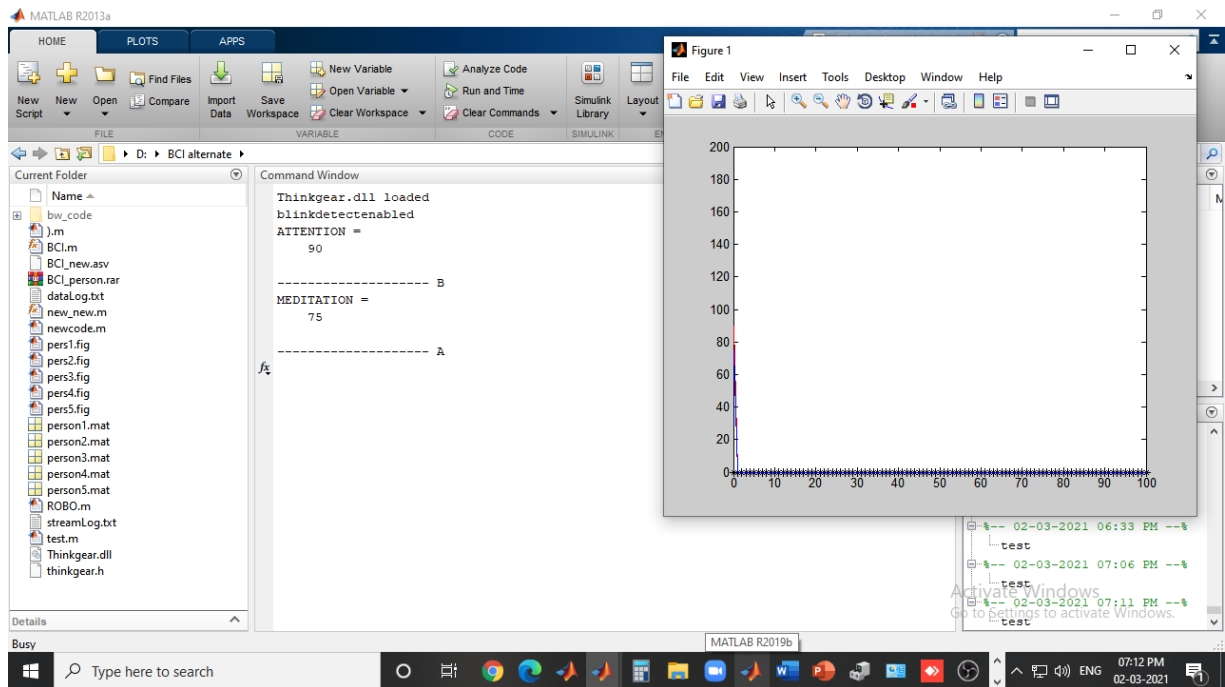
RESULT AND DISCUSSION

5.1 RESULT

we proposed a new experimental for EEG-based robot control, which extended the usage of brain-computer interface. Compared with other paradigms like motor imager and P300, SSVEP is better for real-time application because SSVEP-based paradigm is not a subject-specific BCI, which requires individual data calibration regularly and system training, and it has achieved higher accuracy across subjects. For different subjects, the corresponding conditions are also different. During the experiment, we have selected the appropriate EEG acquisition sensor according to different subjects.

Non-invasive brain-computer interface technology has built a bridge between human brain and smart robots, which has important research significance. In the near future, daily life BCI applications will be involved in fields that are more new. With the development of science and technology, brain-computer interface technology will not only bring hope to people with disabilities, but will also be more integrated into the life of ordinary people, bringing more convenience and use to our life





MEDITATION =
14

BLINK =
42

ATTENTION =
64

Fig 5.1 Input Signal values from Brainwave Sensor



Fig 5.2 Neurosky Mindwave Mobile

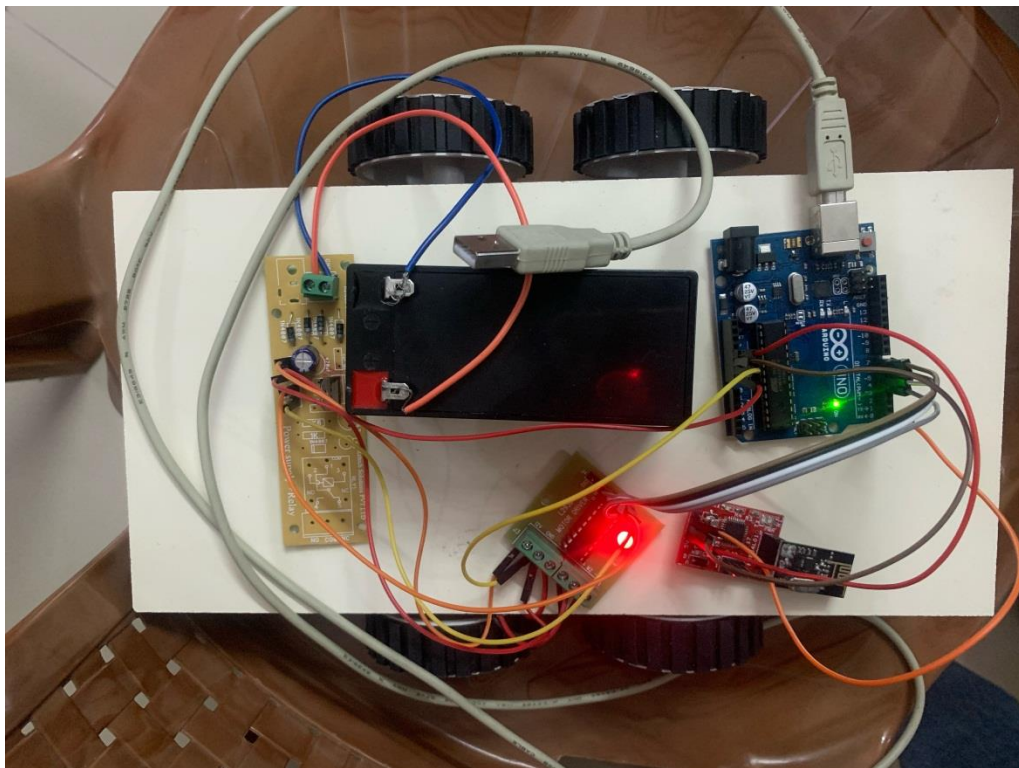


Fig 5.3 Robot

5.2 DISCUSSION

Working of this experiment is getting brain wave signal from the brain using Mindwave sensor those values are shown in MATLAB by running a Programme. Then using Arduino and Zibee the command will be to send to the Robot ,based on that it will move and work. Through this experiment, we hope that in the future, we can recruit more subjects to verify the proposed system and make complex system in different conditions such as controlling a swarm of cleaning robots by one operator brain only.

Advantages of our method are We can even manipulate video games using our minds,Encourage a deaf person to hear by transmitting auditory data to their mind. Allow paralysed individuals to walk about with the help of their minds and Disadvantages are The new technology has not yet been processed ,There may be an incidence of adverse impacts to the atmosphere as a result of using these technologies. By inserting electrodes in a non-invasive way, we get relatively little electric signals. Intrusive placement of electrodes causes scar tissue to form in the brain

5.3 SUMMARY

Robot control based on EEG, increasing the use of the brain-computer interface. SSVEP is better for real-time application than other paradigms such as motor imager and P300 because it is not a subject-specific BCI that needs individual data calibration and device training on a regular basis, and it has achieved greater precision across subjects.The related conditions for different subjects are often different. Throughout the experiment, we selected the appropriate EEG acquisition sensor for each subject.

The design of non-invasive brain-computer interface technology has established a connection between the human brain and smart robots, which has major research implications. In the not-too-distant future, BCI technologies in everyday life can be used in fields that are more novel. With the advancement of science and technology, brain-computer interface technology will not only provide hope to people with disabilities, but will also become more incorporated into everyday life, adding greater comfort and utility to our lives. We hope that by conducting this experiment, we will be able to attract more subjects in the future to test the proposed method and create complex systems in various environments, such as controlling a swarm of cleaning robots with only one operator's brain.

5.4 CONCLUSION:

Brain signals represent the brain's controlled processes and behaviour, as well as the effects of signals obtained from other areas of the body. The scientific group is interested in BCI applications. Several research on the increasing interest in BCI application fields such as medical, transportation, games, and entertainment are discussed in this article.

5.5 Future Work

Using multiple speech recognition modules, we can have simple interactions with smartphones, laptops, and computer games. However, these modules are inefficient. We regulate these consumer electronic products in a variety of ways, such as with a remote control or a gesture.

These brain computer interfaces, also known as the 'mind machine' gui, are the subject of research. This mind-computer interfaces are beneficial in establishing a standard relationship between the mind and the machine. People talk wirelessly using universal translator chips in an intrusive way. Chips are implanted in the brain here.

REFERENCE

1. Alqasemi and R. Dubey, *Advances in Robot Manipulators*. Rijeka, Croatia: InTech, 2010, ch. 3.
2. Bensch, A. A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, W. Rosenstiel, and N. Nesi Birbaumer, "An EEG controlled web browser for severely paralyzed patients," *Comput. Intell. Neurosci.*, vol. 2007, pp. 1–5, 2007.
3. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralyzed," *Nature*, vol. 398, Mar. 1999, pp. 297–298.
4. Choi and A. Cichocki, "Control of a wheelchair by motor imagery in real time," in *Proc. 9th Int. Conf. Intell. Data Eng. Autom. Learning*, 2008, pp. 330–337.
5. Chen, Zhang, Jiang, Y. Zhuang, and X. Zheng, "P300-based brain computer interface: Prototype of a Chinese speller," *J. Comput. Inf. Syst.*, vol. 4, no. 4, pp. 1515–1522, 2008.
6. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: assessing the speed of a P300-based brain–computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 8, no. 2, Jun. 2000, pp. 174–179.
7. Guan, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1285–1294, 2008.
8. Hong, F. Guo, T. Liu, X. Gao, and S. Gao, "N200-speller using motiononset visual response," *Clin. Neurophysiol.*, vol. 120, no. 9, Sep. 2009, pp. 1658– 1666.
9. Karim, T. Hinterberger, and J. Richter, "Neural internet: Web surfing with brain potentials for the completely paralyzed," *Neurorehabil. Neural Repair*, vol. 20, , 2006, no. 4, pp. 508–515.
10. Krepki, B. Blankertz, G. Curio, and K.-R. Muller, "The Berlin brain– " computer interface (BBCI): Towards a new communication channel for online control in gaming applications," *J. Multimedia Tools Appl.*, vol. 33, no. 1, Apr. 2007, pp. 73–90.

11. Millan, F. Renkens, J. Mourino, and W. Gerstner, "Noninvasive ~ brain-actuated control of a mobile robot by human EEG," *IEEE Trans. Bio. Eng.*, vol. 51, no. 6, Jun. 2004, pp. 1026–1033.
12. Millan, R. Rupp, . R. M ¨ uller-Putz, R. Murray-Smith, ¨ . Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kubler, ¨ R. Leeb, C. Neuper, K.-R. Muller, and D. Mattia, "Combining brain– ¨ computer interfaces and assistive technologies state-of-the-art and challenges," *Frontiers Neurosci.*, vol. 4, 2010, , pp.1–15.
13. Muller and B. Blankertz, "Toward noninvasive brain–computer ¨ interfaces," *IEEE Signal Process. Mag.*, vol. 23, no. 5, Sep. 2006, pp. 125–128.
14. Mugler, M. Bensch, S. Halder, W. Rosenstiel, M. Bogdan, N. Birbaumer, and A. Kubler, "Control of an Internet browser using the P300 event-related potential," *Int. J. Bioelectromagnetic*, vol. 10, no. 1, 2008, pp. 56–63.
15. Nijholt, Tan, Allison, J. d. R. Millan, B. Graimann, and ¨ M. M. Jackson, "Brain–computer interfaces for HCI and games," in *Proc. ACM CHI*, Florence, Italy, 2008, pp. 3925–3928.
16. Nijholt, Tan, G. Pfurtscheller, C. Brunner, J. del R. Millan, ¨ B. Allison, B. Graimann, F. Popescu, B. Blankertz, and K.-R. Muller, ¨ "Brain–computer interfacing for intelligent systems," *IEEE Intell. Syst.*, vol. 23, no. 3, May/Jun. 2008, pp. 72–79.
17. Nijholt, Tan, Allison, J. d. R. Millan, B. Graimann, and ¨ M. M. Jackson, "Brain–computer interfaces for HCI and games," in *Proc. ACM CHI*, Florence, Italy, 2008, pp. 3925–3928.
18. Perrin. , "Semi-autonomous navigation of an assistive robot using low throughput interfaces," Ph.D. dissertation, ETHZ, Zurich, Switzerland, 2009.
19. Pires, M. Castelo-Branco, and U. Nunes, "Visual P300-based BCI to steer a wheelchair: A Bayesian approach," in *Proc. IEEE Eng. Med. Biol. Soc.*, 2008, pp. 658–661.
20. Rebsamen. , Guan., Zhang ., Wang ., Teo. , Ang .M. H, Jr., and Burdet ., "A brain controlled wheelchair to navigate in familiar environments," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, Dec. 2010, pp. 590–598.

21. Tangermann, M. Krauledat, K. Grzeska, M. Sagebaum, C. Vidaurre, B. Blankertz, and K.-R. Muller, "Playing pinball with non-invasive BCI," in Proc. 22nd Annu. Conf. Neural Inf. Process. Syst., Vancouver, Canada, 2008, pp. 1641–1648.
22. Tanaka, K. Matsunaga, and H. O. Wang, "Electroencephalogram-based control of an electric wheelchair," IEEE Trans. Robot., vol. 21, no. 4, pp. 762–766, Aug. 2005.
23. Wang, H. Zhang, and C. Guan, "An EEG-based BCI system for 2D cursor control," in Proc. IEEE Int. Joint Conf. Neural Netw., 2008, pp. 2214–2219.
24. Williamson, R. Murray-Smith, B. Blankertz, M. Krauledat, and K.-R. Muller, "Designing for uncertain, asymmetric control: Interaction design for brain-computer interfaces," Int. J. Human-Comput. Stud., vol. 67, no. 10, Oct. 2009, pp. 827–841.
25. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," Electroencephalogr. Clin. Neurophysiol., vol. 78, no. 3, Mar. 1991, pp. 252–259.

APPENDICES

```
%run this function to connect and plot raw EEG data
%make sure to change portnum1 to the appropriate COM port
clc;
clear all
close all
delete(instrfindall);

data_BLINK = zeros(1,256); %preallocate buffer
data_MEDITATION = zeros(1,256);
data_ATTENTION = zeros(1,256);

X = zeros(1,256);
mindsensor = 6; %COM Port of Bluetooth#
comPortName1 = sprintf("\\\\.\\COM%d", mindsensor);
% Baud rate for use with TG_Connect() and TG_SetBaudrate().
TG_BAUD_115200 = 9600;
% Data format for use with TG_Connect() and TG_SetDataFormat().
TG_STREAM_PACKETS = 0;
% Data type that can be requested from TG_GetValue().
TG_DATA_BATTERY      = 0;
TG_DATA_POOR_SIGNAL  = 1;
TG_DATA_ATTENTION    = 2;
TG_DATA_MEDITATION   = 3;
TG_DATA_RAW          = 4;
TG_DATA_DELTA        = 5;
TG_DATA_THETA        = 6;
```

```

TG_DATA_ALPHA1      = 7 ;
TG_DATA_ALPHA2      = 8 ;
TG_DATA_BETA1       = 9 ;
TG_DATA_BETA2       = 10 ;
TG_DATA_GAMMA1      = 11 ;
TG_DATA_GAMMA2      = 12 ;
TG_DATA_BLINK_STRENGTH = 37 ;
TG_DATA_READYZONE   = 38 ;

```

```

%load thinkgear dll

```

```

loadlibrary('Thinkgear.dll');

```

```

fprintf('Thinkgear.dll loaded\n');

```

```

% Get a connection ID handle to ThinkGear

```

```

connectionId1 = calllib('Thinkgear', 'TG_GetNewConnectionId');

```

```

if ( connectionId1 < 0 )

```

```

    error( sprintf( 'ERROR: TG_GetNewConnectionId() returned %d.\n', connectionId1 ) );

```

```

end;

```

```

% Set/open stream (raw bytes) log file for connection

```

```

errCode = calllib('Thinkgear', 'TG_SetStreamLog', connectionId1, 'streamLog.txt' );

```

```

if( errCode < 0 )

```

```

    error( sprintf( 'ERROR: TG_SetStreamLog() returned %d.\n', errCode ) );

```

```

end;

```

```

% Set/open data (ThinkGear values) log file for connection

```

```

errCode = calllib('Thinkgear', 'TG_SetDataLog', connectionId1, 'dataLog.txt' );

```

```

if( errCode < 0 )

```

```

    error( sprintf( 'ERROR: TG_SetDataLog() returned %d.\n', errCode ) );

```

```

end;

```

```

% Attempt to connect the connection ID handle to serial port "COM3"

errCode = calllib('Thinkgear', 'TG_Connect',
connectionId1,comPortName1,TG_BAUD_115200,TG_STREAM_PACKETS );
%if ( errCode < 0 )
    %  error( sprintf( 'ERROR: TG_Connect() returned %d.\n', errCode ) );
%else disp('conected!!');
%end
%%disp('hi');
% fprintf( 'Connect
if(calllib('Thinkgear','TG_EnableBlinkDetection',connectionId1,1)==0)
    disp('blinkdetectenabled');
end
%%
%record data
delete(instrfindall);
serialPortName = 'COM8' ;
blinkCount=0;
attentionCount=0;
meditationCount=0;
Blink=0;
Drive_mode = 0;
count = 0;
X = 0:1:255;
while (i < 50) %loop for 20 seconds

    if (calllib('Thinkgear','TG_ReadPackets',connectionId1,1) == 1) %if a packet was read...

```

```

        if
(calllib('Thinkgear','TG_GetValueStatus',connectionId1,TG_DATA_BLINK_STRENGTH )
~= 0) %if RAW has been updated

        blinkCount = blinkCount + 1;

        data_BLINK
(blinkCount)=calllib('Thinkgear','TG_GetValue',connectionId1,TG_DATA_BLINK_STREN
GTH );

        disp('BLINK = ');
        disp(data_BLINK (blinkCount));

%         if( data_BLINK (blinkCount) <50 && data_BLINK (blinkCount) >100 )
%         disp('----- A')
%         s = serial(serialPortName,'BaudRate',9600);
%         fopen(s);
%         fprintf(s,'A')
%         fwrite(s,'A')
%         fclose(s);
%     end
%     if( data_BLINK (blinkCount) <50 )
%     disp('----- B')
%     s = serial(serialPortName,'BaudRate',9600);
%     fopen(s);
%     fprintf(s,'B')
%     fwrite(s,'B')
%     fclose(s);
% end
%     if( data_BLINK (blinkCount) >100 )
%     disp('----- C')
%     s = serial(serialPortName,'BaudRate',9600);
%

```

```

%         fopen(s);
%         fprintf(s,'C')
%         fwrite(s,'C')
%         fclose(s);
%     end
end

if (calllib('Thinkgear','TG_GetValueStatus',connectionId1,TG_DATA_ATTENTION )
~= 0) %if RAW has been updated
    attentionCount = attentionCount +1;
    data_ATTENTION(attentionCount) =
calllib('Thinkgear','TG_GetValue',connectionId1,TG_DATA_ATTENTION );
    disp('ATTENTION = ');
    disp(data_ATTENTION(attentionCount));
%
%     if( data_ATTENTION (attentionCount) >20)
%         disp('----- B')
% %         s = serial(serialPortName,'BaudRate',9600);
% %
% %         fopen(s);
% %         fprintf(s,'B')
% %         fwrite(s,'B')
% %         fclose(s);
%     end
end

if (calllib('Thinkgear','TG_GetValueStatus',connectionId1,TG_DATA_MEDITATION )
~= 0) %if RAW has been updated
    meditationCount = meditationCount + 1;

```

```

        data_MEDITATION(meditationCount) =
calllib('Thinkgear','TG_GetValue',connectionId1,TG_DATA_MEDITATION );

        disp('MEDITATION = ');

        disp(data_MEDITATION(meditationCount));

%
%      if( data_MEDITATION (meditationCount) >50)
%      disp('----- C')
% %      s = serial(serialPortName,'BaudRate',9600);
% %
% %      fopen(s);
% %      fprintf(s,'C')
% %      fwrite(s,'C')
% %      fclose(s);
%      end
    end

    plot(X,data_ATTENTION,'-r',X,data_BLINK,'-*k',X,data_MEDITATION,'-b');
    axis([0 100 0 200])

    drawnow;

end

delete(instrfindall);

end

%disconnect

calllib('Thinkgear', 'TG_FreeConnection', connectionId1 );

% end

```