**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

Parkinson’s disease (PD) is the most common neurogenerative disease after Alzheimer and it is attacking 1% of the elderly population of above 65 years old in the world. It is triggered by a deficiency of dopamine in basal ganglia, caused by the loss of dopaminergic neurons in substantia nigra pars compacta, a brain part located in the basal plate of *mesenphalon* (midbrain). Some symptoms that are often diagnosed include bradykinesia, resting tremor with a frequency of 4-6 Hz on the hands, muscle rigidity, and an odd walking gait (often called shuffling gait). With the presence of biopotential technology, the muscle activity pattern of Parkinson’s disease patients can be checked and recorded using electromyography (EMG) techniques. Generally, Parkinson’s disease patients’ health information can only be acquired during their visits to clinics or hospitals where they are being treated, whereas the aspects of the disease are also shown during in-between of those visits.

Parkinson's disease is a neurological condition that causes tremors, stiffness, and difficulty in walking, balancing, and coordinating. Signs of the disease generally occur gradually and continue to worsen. People with Parkinson's disease sometimes have difficulties walking and speaking as the disease develops. They might have mental and behavioural changes, as well as sleep issues, depression, memory problems, and exhaustion. Parkinson's disease affects both men and women. However, men are affected by the condition to a larger extent than women. Age is a well-known potential risk for Parkinson's disease. The most people with Parkinson's disease develop the disorder around the age of 60, however about 5-10% of people with Parkinson's disease is developed before the age of 60.

When nerve cells in the basal ganglia, a part of the brain that controls movement, become damaged or die, Parkinson's disease develops. These nerve cells, normally produce dopamine, an important brain neurotransmitter. When neurons die or become damaged, they produce less dopamine, resulting in Parkinson's movement issues. Scientists are indeed unsure what causes dopamine-producing cells to die. Lewy bodies, peculiar clumps of the protein alpha-synuclein, are found in many brain cells of persons with Parkinson's disease. Scientists are striving to learn more about alpha-normal synuclein's and pathological roles, as well as its association to genetic abnormalities that cause Parkinson's disease and Lewy body dementia. It appears to be inherited in certain cases, and a few cases have been linked to specific genetic alterations. While genetics may have a role in Parkinson's disease, the disease does not appear to run in families in most cases. Many scientists now believe that Parkinson's disease is caused by a combination of hereditary and environmental factors, including toxic exposure.

Nerve endings that produce norepinephrine, the major chemical messenger of the sympathetic nervous system, which controls numerous bodily functions such as heart rate and blood pressure, are also lost in people with Parkinson's disease. Some of the non-movement indications of Parkinson's disease, such as weariness, inconsistent blood pressure, decreased food transit through the gastrointestinal tract, and a quick drop in blood pressure when a person rises up from a sitting or laying posture, could be explained by the loss of norepinephrine.

Parkinson's disease symptoms can appear to be mild and grow over time. Affected individuals may experience slight tremors or difficulty in getting out of a couch, for example. They may observe that they speak too softly, or that their handwriting is slow and confined, or that their handwriting is small and cramped. The caretaker notices that the person's face is expressionless and unanimated, or that an arm or leg is not moving normally. People with Parkinson's often develop a parkinsonian gait that includes a tendency to lean forward, small quick steps as if hurrying forward, and reduced swinging of the arms. They also may have trouble initiating or continuing movement.

Detecting and monitoring the motor symptoms plays an important role in early detection of the disease and helps doctors perform early diagnosis. Trying to suggest Parkinson's disease from the presence of a single symptom is unreliable, as the disease develops differently in each patient, and the requirement for active participation may interfere with data gathering protocol adherence.

There are presently no blood or laboratory tests offered to diagnose Parkinson's disease in non-genetic cases. A person's medical history and a neurological examination are usually used to diagnose the disorder. Another sign that the person has Parkinson's is if symptoms improve after starting medication.

Home monitoring can be proposed as one of the solutions to the urgency of Parkinson’s disease symptoms monitoring. Using home monitoring, patients can be checked more frequently, so that their health information are always renewed and up to date. Hopefully, this frequently updated data will improve the patients’ treatments as it is allowing them to be used as a more effective and relevant reference of the treatment. Hereafter, by utilizing EMG techniques, patients’ treatment can be done based on the development of their muscle activity pattern. Home monitoring is allowing these frequent check-ups to be made easier by a wireless and portable EMG device. The pattern characteristics from the acquired EMG signal can be then processed by using discrete wavelet transform (DWT) and feature extraction for further analysis.

Prior to this research, there were also several notable researches regarding the use of home monitoring and remote monitoring methods and devices for Parkinson’s disease patients. The first publication was about a computer-based exercise to support Parkinson’s disease diagnosis and it was first published in 2002. Hereafter, in 2008, a pedometer based wearable step counter was designed using a force sensing resistor, aiming to record expansion activity of the gastronomies muscle—outermost muscle of shank—to see the characteristics of foot movement and to count the number of steps a Parkinson’s disease patient can make during a certain period of time. This research was then followed in 2009 by a design of a tri-axial accelerometer-based gait sensor to measure accelerations carried out from walking by several Parkinson's disease patients and people in good health condition as the subject of research.

The development of remote monitoring has brought it to be able to be done from patients’ homes. This activity is then acknowledged as home monitoring. During 2010 and 2012, two kinds of research regarding this matter emerged. Both of the researches were talking about computer-based motoric examinations.

There was a study of the continuous usage of EMG in 2010. An EMG device was used to detect three main symptoms of Parkinson’s disease, which include bradykinesia, tremor, and muscle rigidity using an accelerometer and dynamometer-based EMG. During the same year, a program was designed to measure accelerations, torque, and how fast could the fingers of a Parkinson’s disease patient tap.

Following the previous researches, this research aims to record and detect the muscle activity of Parkinson’s disease patients using a ESP32-based wireless EMG device. ESP32is a microcontroller equipped with an ESP8266 Wi-Fi module, capable of working from 50 to 70m from its Wi-Fi signal source. This enables the device to be used in different rooms of the same building as long as it is connected to the same server network. The small size of ESP32itself can also be considered an advantage as it requires less dimension and space. The utilization of ESP32also allows the acquired muscle signal to be integrated with the Internet of Things (IoT) system and therefore opens the door to any future possibilities. To make it more user-friendly, the EMG device is designed to be a wearable device.

In this project, we take a first step towards a data-driven approach for the early detection of PD by developing a screening methodology that is based on how an individual interacts with prototype during everyday life. Here we aim to detect PD using a multi-symptom approach that evaluates passively captured data from multiple sensors. The prototype is designed for automated PD detection based on multiple data sources, and covering different PD motor symptoms, that are unobtrusively captured from sensors. Sensors that capture body motions, such as accelerometers, gyroscopes, or electromagnetic motion trackers, are used to analyse motor symptoms of Parkinson's disease. Other types of sensors have also been employed, such as those that measure bioelectrical activity (electromyography).

**1.2 Motivation of the Project**

Parkinson's disease can't be cured, but medications can help control the symptoms, often dramatically. In some more advanced cases, surgery may be advised. The health care provider may also recommend lifestyle changes, especially ongoing aerobic exercise. In some cases, physical therapy that focuses on balance and stretching is important. A speech-language pathologist may help improve speech problems.

Medications may help you manage problems with walking, movement and tremor. These medications increase or substitute for dopamine. People with Parkinson's disease have low brain dopamine concentrations. However, dopamine can't be given directly as it can't enter the brain. You may have significant improvement of your symptoms after beginning Parkinson's disease treatment. Over time, however, the benefits of drugs frequently diminish or become less consistent. You can usually still control your symptoms well.

Medications your health care provider may prescribe include:

**Carbidopa-levodopa.** (Rytary, Sinemet, Duopa, others), Levodopa, the most effective Parkinson's disease medication, is a natural chemical that passes into your brain and is converted to dopamine.

Levodopa is combined with carbidopa (Lodosyn), which protects levodopa from early conversion to dopamine outside your brain. This prevents or lessens side effects such as nausea.

Side effects may include nausea or light-headedness when you stand (orthostatic hypotension).

After years, as your disease progresses, the benefit from levodopa may lessen, with a tendency to wax and wane ("wearing off").

Also, you may experience involuntary movements (dyskinesia) after taking higher doses of levodopa. Your health care provider may lessen your dose or adjust the times of your doses to control these effects.

Unless told otherwise by your health care provider, carbidopa-levodopa is best taken on an empty stomach if you have advanced Parkinson's disease.

**Inhaled carbidopa-levodopa.** Inbrija is a brand-name drug delivering carbidopa-levodopa in an inhaled form. It may be helpful in managing symptoms that arise when oral medications suddenly stop working during the day.

**Carbidopa-levodopa infusion.** Duopa is a brand-name medication combining carbidopa and levodopa. However, it's administered through a feeding tube that delivers the medication in a gel form directly to the small intestine.

Duopa is for patients with more-advanced Parkinson's who still respond to carbidopa-levodopa, but who have a lot of fluctuations in their response. Because Duopa is continually infused, blood levels of the two drugs remain constant.

Placement of the tube requires a small surgical procedure. Risks associated with having the tube include the tube falling out or infections at the infusion site.

**Dopamine agonists.** Unlike levodopa, dopamine agonists don't change into dopamine. Instead, they mimic dopamine effects in your brain.

Dopamine agonists aren't as effective as levodopa in treating symptoms. However, they last longer and may be used with levodopa to smooth the sometimes off-and-on effect of levodopa.

Dopamine agonists include pramipexole (Mirapex ER), and rotigotine (Neupro, given as a patch). Apomorphine (Apokyn) is a short-acting injectable dopamine agonist used for quick relief.

Some of the side effects of dopamine agonists are like the side effects of carbidopa-levodopa. But they can also include hallucinations, sleepiness and compulsive behaviors such as hypersexuality, gambling and eating. If you're taking these medications and you behave in a way that's out of character for you, talk to your health care provider.

**MAO B inhibitors.** These medications include selegiline (Zelapar), rasagiline (Azilect) and safinamide (Xadago). They help prevent the breakdown of brain dopamine by inhibiting the brain enzyme monoamine oxidase B (MAO B). This enzyme metabolizes brain dopamine. Selegiline given with levodopa may help prevent wearing-off.

Side effects of MAO B inhibitors may include headaches, nausea or insomnia. When added to carbidopa-levodopa, these medications increase the risk of hallucinations.

These medications are not often used in combination with most antidepressants or certain narcotics due to potentially serious but rare reactions. Check with your health care provider before taking any additional medications with an MAO B inhibitor.

**Catechol O-methyltransferase (COMT) inhibitors.** Entacapone (Comtan) and opicapone (Ongentys) are the primary medications from this class. This medication mildly prolongs the effect of levodopa therapy by blocking an enzyme that breaks down dopamine.

Side effects, including an increased risk of involuntary movements (dyskinesia), mainly result from an enhanced levodopa effect. Other side effects include diarrhoea, nausea or vomiting.

Tolcapone (Tasmar) is another COMT inhibitor that is rarely prescribed due to a risk of serious liver damage and liver failure.

**Anticholinergics.** These medications were used for many years to help control the tremor associated with Parkinson's disease. Several anticholinergic medications are available, including benztropine (Cogentin) or trihexyphenidyl.

However, their modest benefits are often offset by side effects such as impaired memory, confusion, hallucinations, constipation, dry mouth and impaired urination.

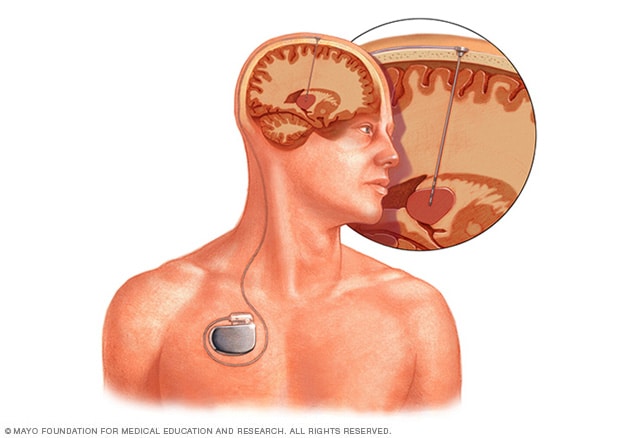
**Amantadine.** Health care providers may prescribe amantadine (Gocovri) alone to provide short-term relief of symptoms of mild, early-stage Parkinson's disease. It may also be given with carbidopa-levodopa therapy during the later stages of Parkinson's disease to control involuntary movements (dyskinesia) induced by carbidopa-levodopa.

Side effects may include a purple mottling of the skin, ankle swelling or hallucinations.

**Adenosine receptor antagonists (A2A receptor antagonist).** These drugs target areas in the brain that regulate the response to dopamine and allow more dopamine to be released. Istradefylline (Nourianz) is one of the A2A antagonist drugs.

**Nuplazid (Pimavanserin).** This drug is used to treat hallucinations and delusions that can occur with Parkinson's disease. Experts aren't sure how it works.

**Surgical procedures**

**Fig 1.1** **Deep brain stimulation Open pop-up dialog box**

**Deep brain stimulation.** In deep brain stimulation (DBS), surgeons implant electrodes into a specific part of the brain. The electrodes are connected to a generator implanted in your chest near your collarbone that sends electrical pulses to your brain and may reduce your Parkinson's disease symptoms.

Your health care provider may adjust your settings as necessary to treat your condition. Surgery involves risks, including infections, strokes or brain haemorrhage. Some people experience problems with the DBS system or have complications due to stimulation. Your doctor may need to adjust or replace some parts of the system.

Deep brain stimulation is most often offered to people with advanced Parkinson's disease who have unstable medication (levodopa) responses. DBS can stabilize medication fluctuations, reduce or halt involuntary movements (dyskinesia), reduce tremor, reduce rigidity, and improve movements.

DBS is effective in controlling changing responses to levodopa or for controlling dyskinesia that doesn't improve with medication adjustments.

However, DBS isn't helpful for problems that don't respond to levodopa therapy apart from a tremor. A tremor may be controlled by DBS even if the tremor isn't very responsive to levodopa.

Although DBS may provide sustained benefit for Parkinson's symptoms, it doesn't keep Parkinson's disease from progressing.

**Advanced treatments**

MRI-guided focused ultrasound (MRgFUS) is a minimally invasive treatment that has helped some people with Parkinson's disease manage tremors. Ultrasound is guided by an MRI to the area in the brain where the tremors start. The ultrasound waves are at a very high temperature and burn areas that are contributing to the tremors.

**Lifestyle and home remedies**

You'll need to work closely with your health care provider to find a Parkinson's treatment plan that offers you the greatest relief from symptoms with the fewest side effects. Certain medicines can make your symptoms worse, so please discuss any drugs you currently take with your health care provider. Certain lifestyle changes may also help make living with Parkinson's disease easier.

**Healthy eating**

While no food or combination of foods has been proved to help in Parkinson's disease, some foods may help ease some of the symptoms. For example, eating foods high in fiber and drinking plenty of fluids can help prevent constipation that is common in Parkinson's disease.

A balanced diet also provides nutrients, such as omega-3 fatty acids, that might be beneficial for people with Parkinson's disease.

**Exercise.**

Exercising may increase your muscle strength, flexibility and balance. Exercise can also improve your well-being and reduce depression or anxiety. Your health care provider may suggest that you work with a physical therapist to learn an exercise program that works for you. You may also try exercises such as walking, swimming, gardening, dancing, water aerobics or stretching. Parkinson’s disease can disturb your sense of balance, making it difficult to walk with your usual gait. Exercise may improve your balance. These suggestions may also help:

* Try not to move too quickly.
* Aim for your heel to strike the floor first when you're walking.
* If you notice yourself shuffling, stop and check your posture. It's best to stand up straight.
* Look in front of you, not directly down, while walking.

**Avoiding falls**

In the later stages of the disease, you may fall more easily. In fact, you may be thrown off balance by just a small push or bump. The following suggestions may help:

* Make a U-turn instead of pivoting your body over your feet.
* Distribute your weight evenly between both feet, and don't lean.
* Avoid carrying things while you walk.
* Avoid walking backward.

**Daily living activities**

Daily living activities — such as dressing, eating, bathing and writing — can be difficult for people with Parkinson's disease. An occupational therapist can show you techniques that make daily life easier.If you are having trouble talking, a speech therapist may help. Many patients with Parkinson's disease have speech difficulties such as a slow, weak voice, trouble with consonants, slurred speech, a low voice that has a monotone with little expression, and inappropriate silences. A speech therapist may be able to help with these problems.

**Alternative medicine**

Supportive therapies can help ease some of the symptoms and complications of Parkinson's disease, such as pain, fatigue and depression. When performed in combination with your treatments, these therapies might improve your quality of life:

**Massage.** Massage therapy can reduce muscle tension and promote relaxation. This therapy, however, is rarely covered by health insurance.

**Tai chi.** An ancient form of Chinese exercise, tai chi employs slow, flowing motions that may improve flexibility, balance and muscle strength. Tai chi may also help prevent falls. Several forms of tai chi are tailored for people of any age or physical condition.

A study showed that tai chi may improve the balance of people with mild to moderate Parkinson's disease more than stretching and resistance training.

**Yoga.** In yoga, gentle stretching movements and poses may increase your flexibility and balance. You may modify most poses to fit your physical abilities.

**Alexander technique.** This technique — which focuses on muscle posture, balance and thinking about how you use muscles — may reduce muscle tension and pain.

**Meditation.** In meditation, you quietly reflect and focus your mind on an idea or image. Meditation may reduce stress and pain and improve your sense of well-being.

**Pet therapy.** Having a dog or cat may increase your flexibility and movement and improve your emotional health.

**Relaxation techniques.** These practices help lower your blood pressure, reduce your heart rate and improve muscle tone.

**Coping and support**

Living with any chronic illness can be difficult, and it's not uncommon to feel angry, depressed or discouraged at times. Parkinson's disease can be profoundly frustrating, as walking, talking and even eating become more difficult and time-consuming. Depression is common in people with Parkinson's disease. But antidepressant medications can help ease the symptoms of depression, so talk with your doctor if you're feeling persistently sad or hopeless.

Although friends and family can be your best allies, the understanding of people who know what you're going through can be especially helpful. Support groups aren't for everyone. However, for many people with Parkinson's disease and their families, a support group can be a good resource for practical information about Parkinson's disease.

Also, groups offer a place for you to find people who are going through similar situations and can support you. Trying to maintain some of your usual activities may be helpful. You aim to do as many things as possible that you could do before the onset of Parkinson's disease.

In rural hospitals, the facilities for health concerned are limited. The lowly quality of health management enables issues in health care system Everyone should get the information of own health as easy and early as possible. Also, it should be worth for each. Newest report of The India Spend analysis of data says that the 500,000 doctors lack in India. WHO defines the doctor patient ratio will be 1:1000 which has been unsuccessful in India. In emerging countries there is lack of resources and management to reach out the problems of individuals. A common man can’t pay for the expensive and daily check-up for his health. For this purpose, various systems which give easy and assured caring unit have been developed. Theses system decreases time with securely handled equipment.

**1.3 Problem Statement**

In existing system, PD is detected at the secondary stage only (Dopamine deficiency) which leads to medical challenges. Also, doctor has to manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge. Thus, the mental disorders are been poorly characterized and have many health complications. PD is generally diagnosed with the following clinical methods as,

* **MRI or CT scan -** Conventional MRI cannot detect early signs of Parkinson's disease
* **PET scan** - is used to assess activity and function of brain regions involved in movement
* **SPECT scan** - can reveal changes in brain chemistry, such as a decrease in dopamine

This results in a high misdiagnosis rate (up to 25% by non-specialists) and many years before diagnosis, people can have the disease. Thus, existing system is not effective in early prediction and accurate medicinal diagnosis to the affected people.

Parkinson's disease (PD) is a chronic neurological illness. It primarily presents as motor symptoms in the initial stages. This stresses the need for an accessible tool that can monitor patients remotely, unobtrusively, and during their daily routine, advising them to see a doctor if it detects motor symptoms that could be linked to the start of Parkinson's disease early. The procedures used to assess and diagnose Parkinson's disease are currently quite subjective, as they typically rely on a clinician's judgement after a series of experiments.

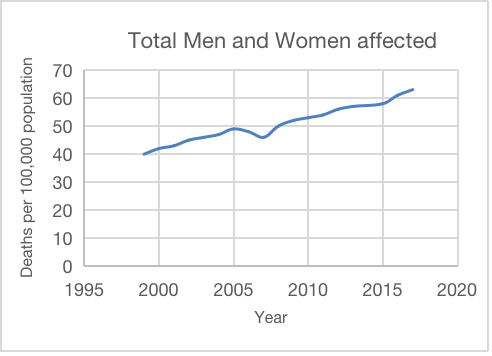
Following the completion of the assessment, a doctor will use a rating scale, such as the Unified Parkinson's Disease Rating Scale (UPDRS), to score the severity of the person's symptoms, such as tremor. The UPDRS includes sections on mentation, behaviour and mood, daily living activities, and motor skills. The motor segment would be employed to check the previously described qualities. The finger tapping test, the Archimedes Spiral test, and the nose tapping test are just a few of the tasks that a person with PD may be asked to do.

In recent years, various interpretations of the Archimedes Spiral test have been established. One of them requires the person with PD to draw straight lines and a circle on a digitizer tablet using a pen/stylus. This type of examination revealed a deficiency that could lead to PD handwriting deficits. This technique of evaluation could help determine a person's level of finger dexterity if they have PD.

A similar assessment tool has been developed based on the conventional Archimedes Spiral test. This method required the user to draw a spiral over a template in the traditional manner. The drawing was subsequently scanned and analysed using a semiautomated computer software. But when the spiral test is done against the PD patient, they eventually get dizzy and fall unconscious in some cases.

Hence, we built a prototype in this project that combines various sensors to detect early-stage symptoms of Parkinson's disease and alert the individual if he is infected by measuring their tremors, body posture, and also monitor their voice by the use of sensors.

The below figure depicts number of people infected with Parkinson’s disease. It is estimated that Parkinson disease has so far affected around 4.5 million individuals worldwide and the number continues to grow.



**Fig 1.2 Total men and women affected by PD**

**1.4 Proposed System**

The monitoring system program that is successfully executed for the case of one normal person’s EMG data in order to ensure that the data can be transferred and well-received via the Wi-Fi platform. The monitoring system trial is carried out in two conditions, namely wired and wireless. In wired condition, the data is successfully displayed on the web page in accordance with the commands on the source code used, which is 4 data/second. However, during the wireless trial, the acquired sampling value is less stable than it is in wired.

Even though the system has successfully displayed 4 data/second during continuous time, at a certain time, it would change to a lower value. Nevertheless, stable or not, the output obtained in these wireless conditions is still very dependent on the internet connection or Wi-Fi that is used for the ESP32. If the router is being accessed by many devices at once, it would affect the performance of the wireless program.

The monitoring system program can also be accessed via a smart phone (Android and IOS based) with the same characteristics as programs accessed through a computer while still connected to the same Wi-Fi network.

**1.5 Objectives**

The main purpose of the paper is to construct a prototype of a spoon which senses the hand motion of an individual suffering from Parkinson’s disease. It gives a counter motion to the trembling actions of a patient’s hand, in order that they don’t spill their food. The movement of the hand acts like feedback to a microcontroller that operates two servos that provide counter-motion. There are Inertial Measurement Units positioned on the spoon that measure the deviations in yaw, pitch, and roll of the spoon and thus govern the counter balancing action of the servos. A ESP32 controller is used to maintain a tilt to the spoon’s angle and as a consequence the spoon is kept at a level plane. Features such as IoT and Machine Learning have been integrated into the system, that enables the product to monitor changes. IoT is used in the device to monitor the sensor reading remotely and analyse the progress of the patient. Machine Learning is used to detect any anomalies in the sensor data from the device. If an anomaly is detected the device send a mail alert to the well-wishers of the patient.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 INTRODUCTION**

Literature survey is an important technique to collect the knowledge on desired topic. By using the information provided in the several literatures give credit to other researchers to prevent duplication. By surveying the several literatures we can identify inconstancies like gaps in research, conflicts in previous studies, open question left from other research. We referred some of the scholars to know some of the information about the project. The scholars which we have referred is listed below.

**2.2 BASE PAPERS**

1. **Xi Wu, Xu Chen, You Duan, Chengjiang Xu, Nan Cheng, Ning An**

**A Study on Gait-based Parkinson’s Disease Detection Using a Force Sensitive Platform**

**2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).**

In this study, we introduce a gait sensing platform that can capture human movement and classify patients with Parkinson’s disease from healthy subjects. Specifically, we first show the platform that consists of force sensitive pressure sensors. Second, we extract gait features from the gait signal collected from the platform. Finally, we collect experimental data from 386 volunteers, including 218 healthy subjects and 168 patients with Parkinson’s disease, and conduct extensive experiments to show the possibility of classifying Parkinson’s disease patients at a high confidence level.

1. **Min Wang1, Bei Wang1, Junhong Zou1, Jian Zhang1 , Masatoshi Nakamura2**

**Hand Movement Compensation for Patients with Parkinson’s Disease Based on Polar Coordination System with Varied Origin**

**2019 4th International Conference on Biomedical Engineering and Informatics (BMEI).**

In this study, a new technique of hand movement compensation for patients with Parkinson’s disease based on spiral drawing tasks was proposed. In order to extract the hand movement feature of patients with Parkinson’s disease, the polar coordination system with varied origin was used to gain characteristic parameters more accurately. The compensation model was constructed by using the parameters obtained in spiral drawing tasks, and we also do simulation under this model to verify the effectiveness.

1. **Benita**

**Detection of Parkinson’s Disease Using Rating Scale**

**2020 International Conference on Computational Performance Evaluation (Compel), North-Eastern Hill University, Shillong, Meghalaya, India. July 2–4, 2020.**

In this study a novel rating scale has been introduced which helps to examine the level of Parkinson Disease but is not mandatory that a person having similar symptoms may surely suffering from Parkinson Disease. PD is an unsolved problem till date hence the study focuses on relevant features, drugs and common techniques used to detect or analyze PD. To overcome such problem different techniques will be used to study and analyze the early detection of PD. It can be analyzed with the help of deep understanding of Parkinson Disease.

1. **Kemal Polat**

**A Hybrid Approach to Parkinson Disease Classification using speech signal: The combination of SMOTE and Random Forests**

**2019 IEEE International Conference.**

In this study was obtained from the UCI machine learning repository. The proposed hybrid machine learning method consists of two stages: i) data pre-processing (oversampling), ii) classification. The Parkinson's disease dataset (PD dataset) is a two-class dataset. While 192 data belong to normal (healthy) individuals, 564 data belong to the diseased class (PD). The data set has an imbalanced class distribution. To transform this imbalanced dataset to balanced dataset, SMOTE (Synthetic Minority Over-Sampling Technique) method is used. Then, after converting to a balanced class distribution, Random Forests classification method was used for classification of Parkinson's disease dataset.

1. **A.M. Arid Handojoseno1, Student Member, IEEE, Moran Gilat2, Quynh Tran Ly1, Hayat Chamtie2, James M. Shine3, Tuan N. Nguyen1, Member IEEE, Yvonne Tran1,4, Simon J.G. Lewis2, Hung T. Nguyen1, Senior Member, IEEE**

**An EEG Study of Turning Freeze in Parkinson’s Disease Patients: The Alteration of Brain Dynamic on the Motor and Visual Cortex**

**In Proc. 31st Annual. Int. Conf. IEEE Eng. Med. Biol. Soc., Chicago, 2017.**

Freezing of gait is a very debilitating symptom affecting many patients with Parkinson’s disease, leading to a reduced mobility and increased risk for falls. Turning is known to be the most provocative trigger for freezing of gait. However, the underlying brain dynamic changes associated with a turning freeze remain unknown. This study therefore used ambulatory EEG to investigate the brain dynamic changes associated with freezing of gait during turning. In addition, this study aimed to determine the most suitable EEG sensor location to detect freezing of gait during turning using our classification system.

1. **Jorge Cancela, Samanta Villanueva Mascato, Dimitrios Gatsios, George Rigas, Andrea Marcante, Giovanni Gentile, Roberta Biundo, Manuela Giglio, Maria Chondrogiorgi, Robert Vilzmann, Spyros Konitsiotis, Angelo Antonini, Maria T. Arredondo and Dimitrios I. Fotiadis**

**Monitoring of motor and non-motor symptoms of Parkinson’s disease**

**In Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, 2018.**

In this work we propose a mHealth approach based on a set of unobtrusive, simple-in-use, off-the-self, co-operative, mobile devices that will be used for motor and non-motor symptoms monitoring and evaluation, as well as for the detection of fluctuations along with their duration through a waking day. Ideally, a multidisciplinary and integrated care approach involving several professionals working together (neurologists, physiotherapists, psychologists and nutritionists) could provide a holistic management of the disease increasing the patient’s independence and Quality of Life (QoL). To address these needs we describe also an ecosystem for the management of both motor and non-motor symptoms on PD facilitating the collaboration of health professionals and empowering the patients to self-manage their condition

1. **A. Bermeo, M. Bravo, M. Huerta, A. Soto**

**A System to Monitor Tremors in Patients with Parkinson's Disease**

**2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).**

In this work the design and development of a system to monitor tremors in patients with Parkinson's disease based on Arduino open-source prototyping platform is presented. For processing data tremors acquired by the sensor device we have developed an Android application which allows an evaluation of the state of PD patients based on three types of tests that are in the Unified Parkinson’s Disease Rating Scale recommended by the Movement Disorder Society (MDSUPDRS); the tests performed in the application are: postural tremor of the hands, kinetic tremors of the hands and resting tremor amplitude. The results of PD Patients showed that despite receiving medication to minimize symptoms of their disease, patients have a considerable tremor amplitude, which affects the normal development of their daily activities.

1. **Nisrina Firyal Fadhlannisa, Basari**

**Design of Wireless Electromyography (EMG) Monitoring System for Muscle Activity Detection on Parkinson Disease**

**2020 International Conference on Smart Technology and Applications (ICoSTA)**

In this research, we develop an Electromyography (EMG) based remote-monitoring systemin order for patients are able to be checked more frequently and the results are in always-renewed information. This monitoring system utilizes NodeMCU as the compiler component and ESP8266 Wi-Fi module and web server. The data acquired from an arm of a tested person will always be read in real-time and can be displayed on web-based platform. From the data acquired, they are processed in detail for pattern characteristics analysis.

1. **L. Cunningham, C. Nugent, G. Moore, D. Finlay, and D. Craig**

**Identifying Fine Movement Difficulties in Parkinson's Disease Using a Computer Assessment Tool**

**Proceedings of the 9th International Conference on Information Technology and Applications in Biomedicine, ITAB 2019, Larnaca, Cyrus, 5-7 November 2009**

A computer-based assessment tool has been created and an evaluation carried out by 10 participants with Parkinson's disease and a control group of 10 participants without the disease. This assessment tool collects data on the user's mouse movements by asking them to click targets on screen, to allow for identification of certain features of Parkinson's disease. Results showed that the level of hand control and finger dexterity was considerably less in the participants with the disease compared to those without.

1. **Niya Romy Markose, Priscilla Dinkar Moyya, Mythili Asaithambi Analysis of Tremors in Parkinson’s Disease Using Accelerometer**

**2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)**

This prototype was designed to observe and quantify the tremor signal from Parkinson's disease patients. The prototype is based on Arduino Uno programming and interfacing, and the ADXL335 tri-axial accelerometer is used as a sensor. The resting tremor signal was acquired in the form of acceleration using the sensor accelerometer from fingertip, wrist and forearm of the patient. The Arduino processed the data which was transferred to MATLAB for further processing. The resting tremor was observed in terms of amplitude and spectral density.

1. **Arash Salarian, Member, IEEE, Heike Russmann, François J. G. Vingerhoets, Pierre R. Burkhard, and Kamiar Aminian, Member, IEEE**

**Ambulatory Monitoring of Physical Activities in Patients With Parkinson’s Disease**

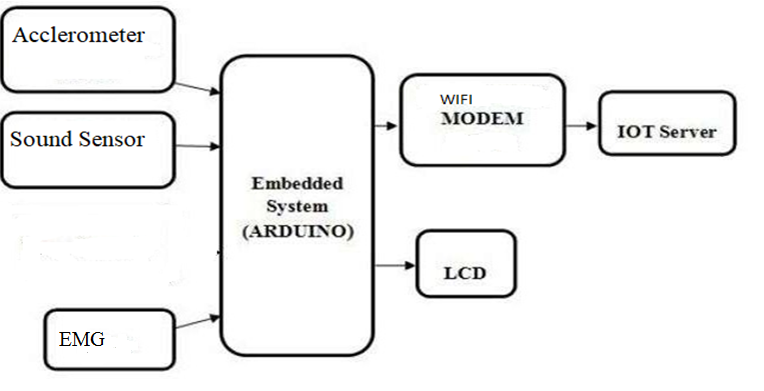
**IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 54, NO. 12, DECEMBER 2017**

A new ambulatory method of monitoring physical activities in Parkinson’s disease (PD) patients is proposed based on a portable datalogger with three body-fixed inertial sensors. A group of ten PD patients treated with subthalamic nucleus deep brain stimulation (STN-DBS) and ten normal control subjects followed a protocol of typical daily activities and the whole period of the measurement was recorded by video. Walking periods were recognized using two sensors on shanks and lying periods were detected using a sensor on trunk. By calculating kinematics features of the trunk movements during the transitions between sitting and standing postures and using a statistical classifier, sit-to-stand (SiSt) and stand-to-sit (StSi) transitions were detected and separated from other body movements.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Block Diagram**

****

**Fig 3.1 Block Diagram**

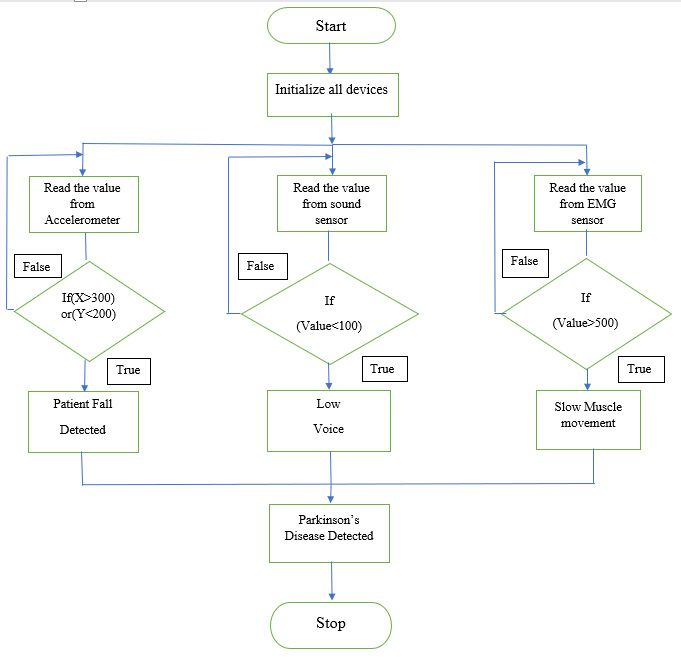
**3.2 Methodology**

The monitoring system program that is successfully executed for the case of one normal person’s EMG data in order to ensure that the data can be transferred and well-received via the Wi-Fi platform. The monitoring system trial is carried out in two conditions, namely wired and wireless. In wired condition, the data is successfully displayed on the web page in accordance with the commands on the source code used, which is 4 data/second. However, during the wireless trial, the acquired sampling value is less stable than it is in wired.

Even though the system has successfully displayed 4 data/second during continuous time, at a certain time, it would change to a lower value. Nevertheless, stable or not, the output obtained in these wireless conditions is still very dependent on the internet connection or Wi-Fi that is used for the ESP32. If the router is being accessed by many devices at once, it would affect the performance of the wireless program.

The monitoring system program can also be accessed via a smart phone (Android and IOS based) with the same characteristics as programs accessed through a computer while still connected to the same Wi-Fi network.

First the person is made to do some simple exercises. These exercises include body movement exercises, speaking exercise, writing, running and walking. The tri-axial accelerometer is used to measure tremors and the data is sent to the microcontroller. Tremor is the involuntary shaking of body parts, and it is one of the most observable symptoms. Sound sensor is used for automatic classification of Parkinson’s disease (PD) speakers and healthy controls (HC) is performed considering speech recordings collected in non-controlled noise conditions. Accelerometer and gyroscope sensors are embedded in the device to get the result of fall detection more accurately. EMG Sensor is used to measure small electrical signals generated by our muscles when you move them. This includes lifting your arm up or clenching your fist. All the data collected from these sensors are sent to the microcontroller for determining whether the person has Parkinson’s disease or not. The sensor values are monitored by the microcontroller. If the sensor values exceed the threshold values, then the information is sent to the doctors and respective caretakers. The device is integrated with IoT framework for patient monitoring, storage of sensor values and cloud access by caregiver for further treatment.



**Fig 3.2 Flow Chart**

The above fig shows the flow chart in which the device works and communicates. Firstly, all the devices are initiated in the loop; the loop leads to three different functions where the process of the sensors used are done. Then in the first function, the accelerometer reads the values and compares it to the threshold value given, if the condition is true PD is detected and the message appears on the LCD display as well as the caretaker’s device. Similarly, the sound sensor and the EMG sensor used in the prototype reads the values and compares it to its respective threshold values to determine the condition is true or not. If the condition appears to be true, it displays the respective messages on both LCD and the caretaker’s device. Then the process stops. If at all the condition is false, the function returns and the person’s health is alright and isn’t diagnosed with PD.

**3.3 System Requirements**

**Hardware Requirements:**

1. Arduino UNO
2. EMG Sensor
3. Accelerometer Sensor
4. Sound Sensor
5. Wi-Fi module
6. LCD Display

**Software Requirements:**

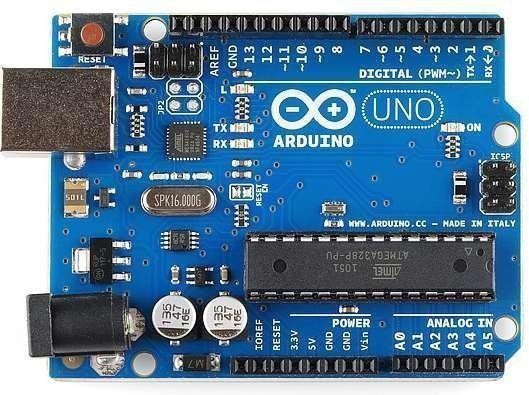
1. Arduino IDE
2. Embedded C

**3.4 Hardware Requirements:**

1. **ARDUINO UNO:**

* Arduino Uno is a microcontroller board based on the ATmega328P.
* It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.
* It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Arduino interface boards provide the engineers, artists, designers, hobbyists and anyone who tinker with technology with a low-cost, easy-to-use technology to create their creative, interactive objects, useful projects etc., A whole new breed of projects can now be built that can be controlled from a computer.

****

**Fig 3.3 Arduino Uno**

* Arduino Uno is a microcontroller board based on the ATmega328P.
* It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.
* It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Arduino is a open source electronics prototyping platform based on flexible, easy-to- use hardware and software. It‘s intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. It‘s an open-source physical computing platform based on a microcontroller board, and a development environment for writing software for the board.

|  |  |
| --- | --- |
| **Microcontroller** | ATmega328P |
| **Operating Voltage** | 5V |
| **Input Voltage** | 7-12V |
| **Input Voltage limit** | 6-20V |
| **Digital I/O pins** | 6 |
| **Analog Input Pins** | 6 |
| **DC current per I/O pin** | 20Ma |
| **Dc current for 3.3V pin** | 50Ma |
| **Flash memory** | of which 0.5KB is used |
| **SRAM** | 2 KB |
| **EEPROM** | 1 KB |
| **Clock speed** | 16MHz |
| **Length** | 68.6mm |
| **Width** | 53.4mm |
| **Weight** | 25g |

**Table 3.1 Arduino specifications**

In simple words, Arduino is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics, such as motors, sensors, laser diodes, loudspeakers, microphones, etc. They can either be powered through the USB connection from the computer or from a9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently

Anyone can buy this device through online auction site or search engine. Since the Arduino is an open-source hardware designs and create their own clones of the Arduino and sell them, so the market for the boards is competitive. An official Arduino costs about $30, and a clone often less than $20.

The name ―Arduino is reserved by the original makers. However, clone Arduino designs often have the letters ―Arduino on the end of their name, for example, Freeduino or DFRduino. The software for programming your Arduino is easy to use and also freely available for Windows, Mac, and LINUX computers at no cost.

**Microcontroller**

Microcontroller can be described as a computer embedded on a rather small circuit board. To describe the function of a microcontroller more precisely, it is a single chip that can perform various calculations and tasks, and send/receive signals from other devices via the available pins. Precisely what tasks and communication with the world it does, is what is governed by what instructions we give to the Microcontroller. It is this job of telling the chip what to do, is what we refer to as programming on it.

However, the uC by itself cannot accomplish much; it needs several external inputs: power, for one; a steady clock signal, for another. Also, the job of programming it has to be accomplished by an external circuit. So typically, a uC is used along with a circuit which provides these things to it; this combination is called a microcontroller board. The Arduino Uno that you have receives, is one such microcontroller board. The actual microcontroller at its heart is the chip called Atmega328. The advantages that Arduino offers over other microcontroller boards are largely in terms of reliability of the circuit hardware as well as the ease of programming and using it.

**Open-source Hardware**

Open-source hardware shares much of the principles and approach of free and open- source software. The founders of Arduino wanted people to study their hardware, to understand how it works, make changes to it and share those changes with the world. To facilitate this, they release all of the original design files (Eagle CAD)for the Arduino hardware. These files are licensed under a Creative Common Attribution Share-Alike license, which allows for both personal and commercial derivative works, as long as they (people) credit Arduino and release their designs under the same license.

The Arduino software is also open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL

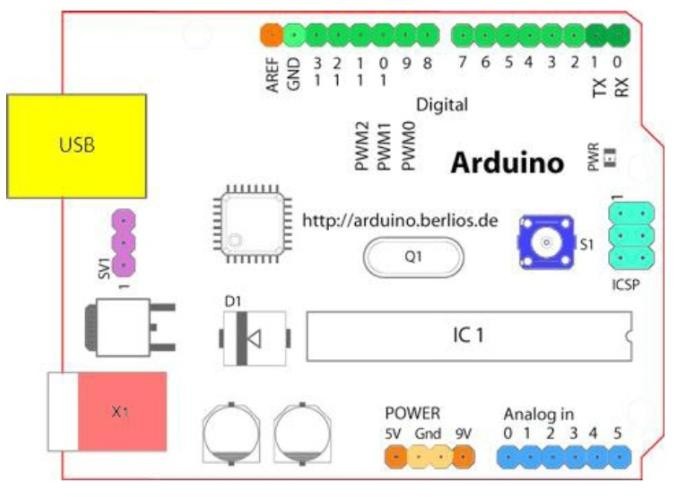
**History of ARDUINO**

While teaching a physical computing class at the Interaction Design Institute Ivrea in 2005, Massimo Banzi’s students were unwilling to spend the 76 euros for the BASIC Stamp microcontrollers commonly used in such applications. Banzi and his colleagues looked for alternatives, finally settling on the *wiring* platform developed by one of Banzi’s students. In his own words:

We started to figure out how we could make the whole platform even simpler, even cheaper, even easier to use. And then we started to essentially re-implement the whole thing as an open source project. Once they had a prototype, a student wrote the software that would allow *wiring* programs to run on the new platform. Upon seeing the project, visiting professor Casey Reas suggested that there might be wider applications than just design schools for the new product. The prototype was redesigned for mass production and a test run of 200 boards was made. Orders began coming in from other design schools and the students looking for Arduinos, and the Arduino project was born and Massimo Banziand David Cuartiellesbecame its founders. ARUDINO‖ is an Italian word, meaning ―STRONG FRIEND. The English version of the name is ―Hardwin. As of May 2011, more than 300,000 Arduino units are ―in the wild.

**Design Goals**

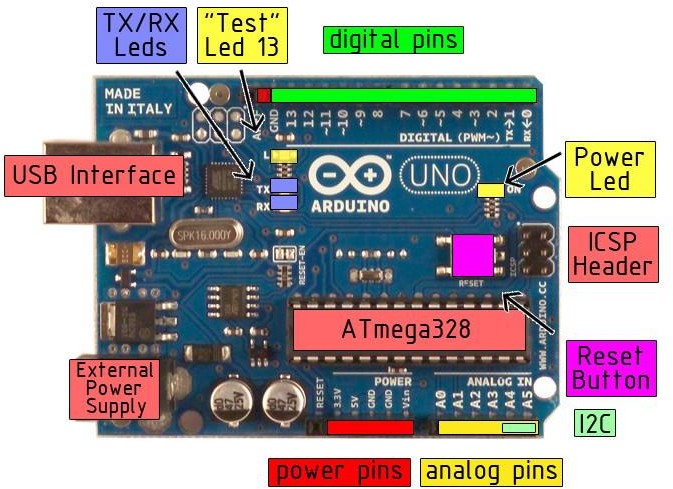
* Work with a Mac (as most design students use one)
* USB connectivity (MacBooks don‘t have serial ports)
* Cheap (about 20 euros)
* More powerful than a BASIC stamp
* Something you could build/fix yourself
* Simple and easy to use by someone without formal electronics training



**Fig 3.4 ARDUINO Board layout**

**Arduino programming**

* The Arduino/Genuino Uno can be programmed with the Arduino Software IDE .Select "Arduino/Genuino Uno from the Tools > Board menu according to the microcontroller on board. The ATmega 328 on the Arduino/Genuino Uno comes pre-programmed with a boot loader that allows us to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol reference, C header files.
* We can also bypass the boot loader and program the microcontroller through the ICSP In- Circuit Serial Programming header using Arduino ISP or similar. The ATmega16U2/8U2 is loaded with a DFU boot loader.
* On Rev1 boards connecting the solder jumper on the back of the board near the map of Italy and then reseing the 8U2.
* On Rev2 or later boards there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

**Arduino Pin Description**

**Fig 3.5** **Arduino Pin Description**

* 16 MHz
* 8 Kbyte Flash RAM(1K taken by the boot loader)
* 1 Kbyte RAM(eg.for auto/local variables and stack)
* 14 digital Input/Output Port
* Single chip USB to async. Serial data transfer interface
* USB 2.0 compatible.
* Transmit and receive LED Drive signals.
* 256 Byte receive, 128 Byte transmit buffer.
* Data transfer rate from 300bits/sec to 2 Mb/sec.

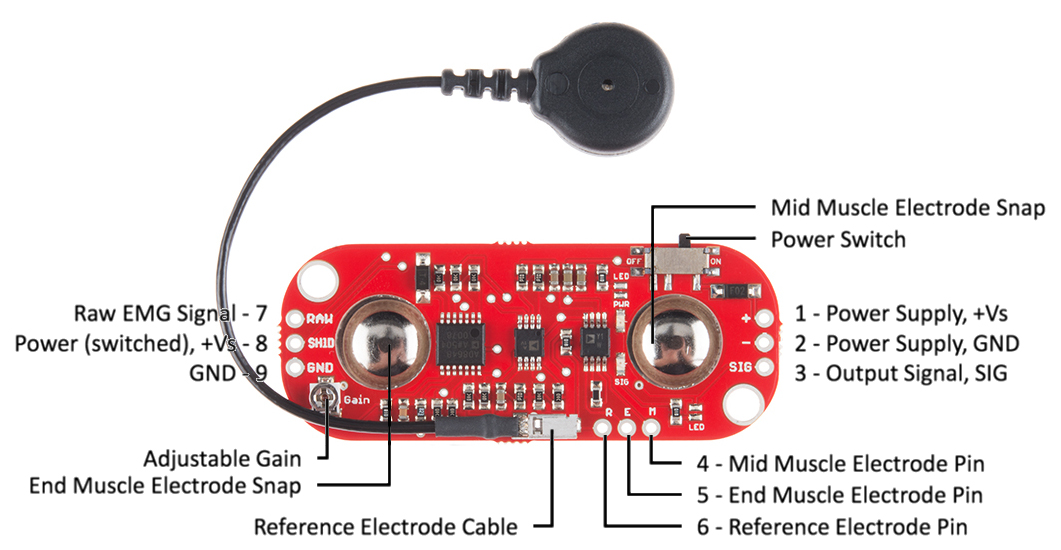
The power requirement for ARDUINO is 9 to 12V DC, 250mA or more, 2.1mm plug, centre pin positive.

**The OFF-the shelf adapter**

* must be a DC adapter (i.e. it has to put out DC, not AC)
* should be between 9V and 12V DC
* must be rated for a minimum of 250mA current output, although you will likely want something more like 500mA or 1A output, as it gives you the current necessary to power a servo or twenty LEDs if you want to.
* must have a 2.1mm power plug on the Arduino end, and
* the plug must be "centre positive", that is, the middle pin of the plug has to be the + connection.
* **Current rating:** Since you'll probably be connecting other things to the Arduino (LEDs, LCDs, servos) you should get an adapter that can supply at least 500mA, or even 1000 mA (1 ampere). That way you can be sure you have enough juice to make each component of the circuit function reliably.
* The Arduinos’ on-board regulator can actually handle up to 20V or more, so you can actually use an adapter that puts out 20V DC. The reasons you don't want to do that are twofold: you'll lose most of that voltage in heat, which is terribly inefficient.
* Secondly, the 9V pin on the Arduino board will actually be putting out 20V or so, which could lead to potential disaster when you connect something expensive to what you thought was the 9V pin. Our advice is to stick with the 9V or 12V DC adapter.

1. **EMG SENSOR:**

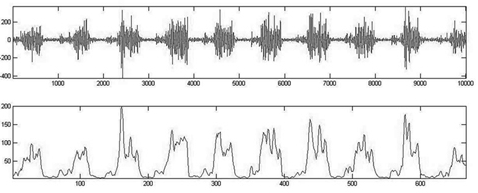
* The process begins with EMG sensor placement; where it’s placed in the innervations zone of both tendons for better detection quality .
* Electrodes begin to detect electrical activity generated by muscle movement/contraction
* Electrical activity detected is then displayed via the form of waves on a monitor (also known as an oscilloscope).



**Fig 3.6 EMG Sensor**

Electromyography (EMG) measures the electrical impulses of the muscles within the body. It provides a quantifiable way to view the activity of a muscle while at rest and throughout the entire range of motion of a movement. The two different EMG techniques are intramuscular and surface EMG. Intramuscular EMG involves an electrode needle being inserted into the muscle tissue to target specific motor units. The muscle should behave in a certain manner in reaction to the needle, and results are compared to that. On the other hand, surface EMG (SEMG) is a noninvasive technique that allows EMG data to be measured without penetrating the skin and provides for a much broader evaluation of a muscle. SEMG picks up the electrical signals that are fired from a population of motor units within a muscle. These electrical signals travel through tissue until they eventually reach the surface of the skin, where electrodes then sense the energy. SEMG is a much more appropriate technique for sampling users in a workout environment and the data gathered will be more appropriate because it relates to a wider range of muscle use.

Once data is gathered from the electrodes, it must be processed before it becomes easily understandable. The initial type of processing is filtering, or removing the unnecessary data from the sample. This attempts to remove the electrical noise that is caused by anything other than the electrical impulses in the muscle. Once the data is properly filtered, the data can either be analyzed in its raw form or it can be further processed. Displaying the raw EMG data in a graph creates an oscillating line, which contains both positive and negative values as shown in Figure 3. From this graph, a user can see when the muscles are activated by looking at the thickness and height of the line. Although this may allow users to get a quick look at their muscle's energy expenditure, it



**Fig 3.7 Raw EMG data (top) and its Root Mean Square (bottom)**

may be harder to draw conclusions from the data without further processing it. To make the data easier to view, all processing techniques first rectify and smooth the data. Rectifying converts the negative electrical potential and adds it to the positive potential, making all values positive. Smoothing the data is done by integral averaging, which involves averaging a set of points and plotting that average for each point instead of plotting each individual point. This reduces variability in the data because now outlying values are averaged with their surrounding values to form a smoother graph.

Aside from processing raw EMG data for display, several processing techniques are also used to quantify the data, yielding numbers that can more easily describe muscle energy expenditure. Peak-to-peak measuring calculates the difference between the top and bottom of each trace and averages this value over a period of time. Integral averaging, the same method used for smoothing EMG graphs, can be calculated and represents .637 of one half of the peak-to-peak value. Root Mean Square (RMS) is a method that squares the data, calculates the average, and then calculates the square-root of this value. RMS is more commonly used than integral averaging because it provides less distortion [26]. This processed EMG data can now be used to evaluate criteria such as:

* + - The activation timing of a muscle: when energy expenditure of a muscle begins and ends and how frequently that occurs within a set period of time.
    - The symmetrical expenditure of muscles: whether symmetrical muscles, such as the left and right bicep, display the same muscle activity levels for an exercise.
    - A fatigue analysis on the muscle: how quickly or slowly the muscle decreases in energy expenditure.

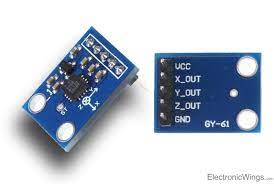
The Shimmer EMG sensor is an attachment to the main Shimmer board and maintains the lightweight and small form factor of the unit. The EMG attachment is a surface EMG that connects to the skin via disposable electrodes and captures the activity of the entire muscle. The data gathered is filtered and the integral average is calculated before storing the data\

1. **ACCELEROMETER SENSOR:**

* The accelerometer sensors measure the acceleration by measuring the change in capacitance.
* Its structure has a mass attached to a spring which moves along one direction and has fixed outer plate so, when acceleration is applied in any direction, the capacitance between the plates and the mass will change.
* The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.
* The user selects the bandwidth of the Accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

**ADXL335 Three Axis Low-g Micro machined Accelerometer**

* The user selects the bandwidth of the Accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.
* The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP\_LQ).



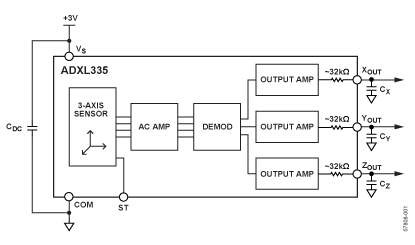
**Fig 3.8 Accelerometer Sensor**

**Features**

* 3-axis sensing
* Small4 mm × 4 mm × 1.45 mm LFCSP
* Low power - 350 μA (typical)
* Single-supply operation1.8 V to 3.6 V
* 10,000 g shock survival
* Excellent temperature stability
* BW adjustment with a single capacitor per axis
* ROHS/WEEE lead-free compliant

**Theory of Operation**

* The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3 g mini-mum. It contains a polysilicon surface-micro machined sensor and signal conditioning circuitry to implement open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.
* The sensor is a polysilicon surface-micro machined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass.



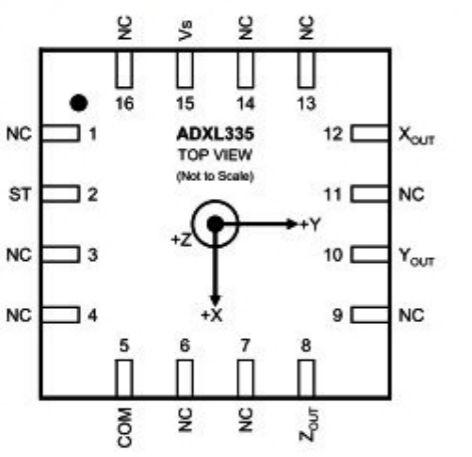
**Fig 3.9 Functional Block diagram**

Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

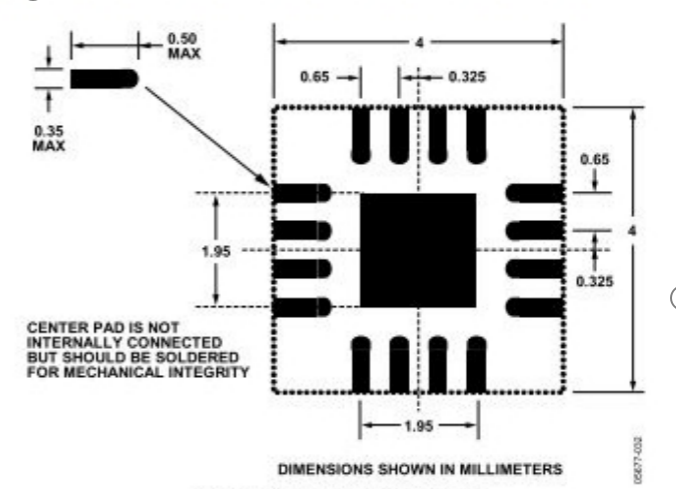
**Applications**

* Motion- and tilt-sensing applications
* Mobile devices
* Gaming systems
* Image stabilization
* Sports and Health Devices

**Pin Diagram**



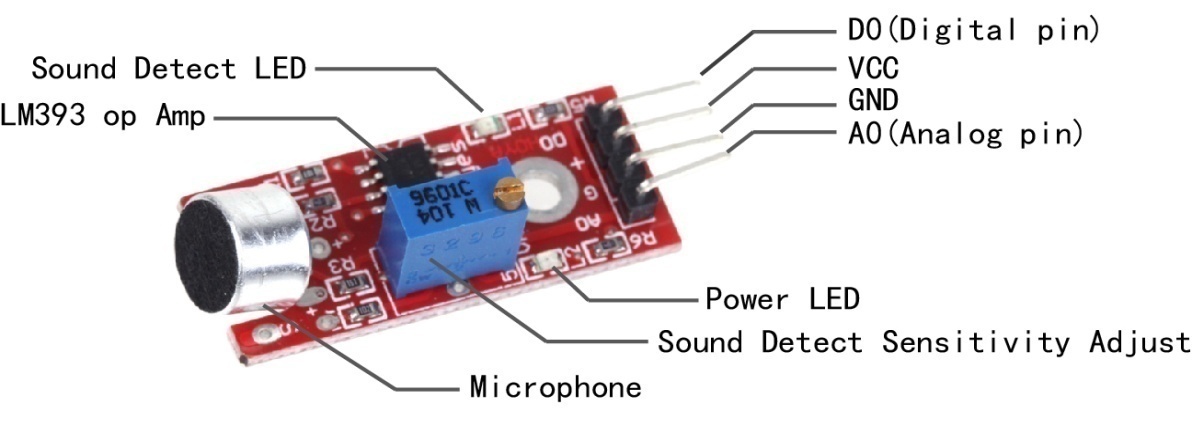
**Fig 3.10 Pin configuration**



**Fig 3.11 Recommended PCB layout**

1. **SOUND SENSOR:**

* The sound sensor has a thin piece of material called a diaphragm that vibrates when hit by sound waves (similar to how your eardrum vibrates when hearing sound).
* The vibration of the diaphragm is converted by the sensor into an electrical signal that is sent to the LEGO brick, which knows that a sound has been heard.



**Fig 3.12 Sound Sensor**

This module can store 15 pieces of voice instruction. Those 15 pieces are divided into 3 groups, with 5 in one group. First we should record the voice instructions group by group. After that, we should import one group by serial command before it could recognize the 5 voice instructions within that group. If we need to implement instructions in other groups, we should import the group first. This Module is speaker-dependent. If your friend speaks the voice instruction instead of you, it may not identify the instruction.

**SERIAL COMMANDS TO OPERATE**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Key***  ***(HEX format)*** | ***Description*** | ***Respond in Common Mode*** | ***Respond in Compact***  ***Mode*** |
| ***0x00*** | *Enter into “Waiting” state* | *"Waiting! \ n" : successful*  *"ERROR! \ n" : Instruction error* | *0xcc :*  *Successful* |
| ***0x01*** | *Delete the instructions of*  *group 1* | *"Group1 Deleted ! \ n" : successful*  *"ERROR! \ n" : Instruction error* | *0xcc :*  *Successful* |
| ***0x02*** | *Delete the instructions of*  *group 2* | *"Group2 Deleted ! \ n" : successful*  *"ERROR! \ n" : Instruction error* | *0xcc :*  *Successful* |
| ***0x03*** | *Delete the instructions of group 3* | *"Group3 Deleted ! \ n" : successful*  *"ERROR! \ n" : Instruction error* | *0xcc :*  *Successful* |
| ***0x04*** | *Delete the instructions of*  *all the 3 groups* | *" All Groups Deleted !\n " : successful*  *"ERROR! \ n" : Instruction error* | *0xcc :*  *Successful* |
| ***0x11*** | *Begin to record instructions of group 1* | *"ERROR! \ n" : Instruction error*  *"START \ n" : Ready for recording, you can speak* | *0xe0 : Instruction error* |

**Table 3.2 Serial commands to operate**

**Advantages**

• Easy to use.

• 15 independent channels can be used.

• Consumes Less Voltage

• Analogue and Digital Output

**Applications**

• Speech controlled appliances and toys

• Speech assisted computer games

Voice Recognition Circuit DAVR001

• Speech assisted virtual reality

• Telephone assistance systems

• Voice recognition security

• Speech to speech translation

1. **WI-FI MODULE:**

* The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network.
* The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.



**Fig 3.13 Wi-Fi Module**

* NodeMCUDev Kit has Arduino like Analog (i.e. A0) and Digital (D0-D8) pins on its board.
* It supports serial communication protocols i.e. UART, SPI, I2C etc.
* Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

**Pin Definition**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **GPIO** | **Input** | **Output** | **Notes** |
| **D0** | **GPIO16** | no interrupt | no PWM or I2C support | HIGH at boot used to wake up from deep sleep |
| **D1** | **GPIO5** | OK | OK | often used as SCL (I2C) |
| **D2** | **GPIO4** | OK | OK | often used as SDA (I2C) |
| **D3** | **GPIO0** | pulled up | OK | connected to FLASH button, boot fails if pulled LOW |
| **D4** | **GPIO2** | pulled up | OK | HIGH at boot connected to on-board LED, boot fails if pulled LOW |
| **D5** | **GPIO14** | OK | OK | SPI (SCLK) |
| **D6** | **GPIO12** | OK | OK | SPI (MISO) |
| **D7** | **GPIO13** | OK | OK | SPI (MOSI) |
| **D8** | **GPIO15** | pulled to GND | OK | SPI (CS) Boot fails if pulled HIGH |
| **RX** | **GPIO3** | OK | RX pin | HIGH at boot |
| **TX** | **GPIO1** | TX pin | OK | HIGH at boot debug output at boot, boot fails if pulled LOW |
| **A0** | **ADC0** | Analog Input | X |  |

**Table 3.3 Pin definition**

**How to start with NodeMCU?**

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols.

To get start with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There is online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

**NodeMCU:**

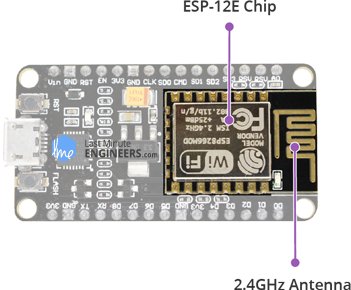
The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized Wi-Fi enabled microcontroller – **ESP8266**, at an unbelievable price! For less than $3, it can monitor and control things from anywhere in the world – **perfect for just about any IoT project**.

## ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having **Tensilica Xtensa® 32-bit LX106 RISC microprocessor** which operates at **80 to 160 MHz** adjustable clock frequency and supports **RTOS**.

ESP-12E Chip

* Tensilica Xtensa® 32-bit LX106
* 80 to 160 MHz Clock Freq.
* 128kB internal RAM
* 4MB external flash
* 802.11b/g/n Wi-Fi transceiver



**Fig 3.14 Wi-Fi Module chip**

There’s also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a Wi-Fi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

**Power Requirement**

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labelled as 3V3. This pin can be used to supply power to external components.

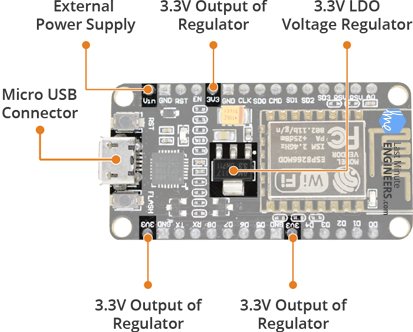
**Power Requirement**

**Operating Voltage:** 2.5V to 3.6V

On-board 3.3V 600mA regulator

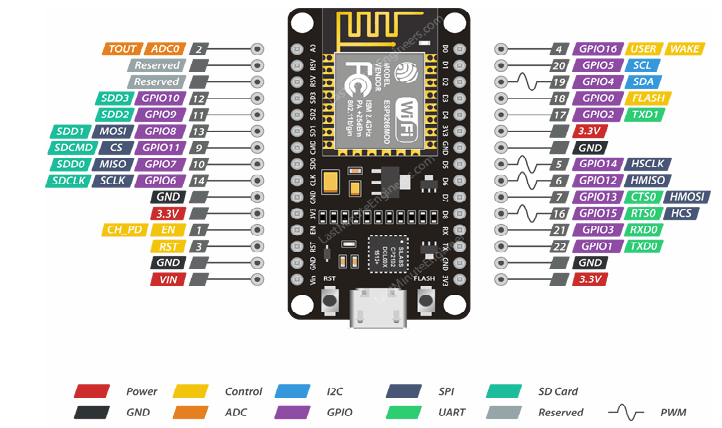
80mA Operating Current

20 µA during Sleep Mode



**Fig 3.15 Wi-Fi module Description**

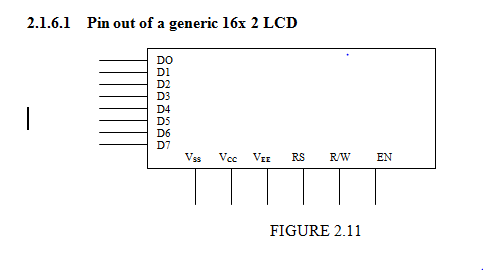
**Power to the ESP8266 NodeMCU**is supplied via the **on-board MicroB USB connector**. Alternatively, if you have a regulated 5V voltage source, the **VIN pin** can be used to directly supply the ESP8266 and its peripherals.

****

**Fig 3.16 Wi-Fi module Pin description**

1. **LCD DISPLAY (16X2)** :

* A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals.
* Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose Wireless Sensors Network for Environmental Radiation Monitoring using IoT ECE 2021-2022 13 computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock.
* They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.
* LCD is used in wide range application including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage.
* Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks.



**Fig 3.17 Pin out of generic 16X2 LCD**

* LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big- screen television sets.
* This LCD can be used to display 16 characters in 2 rows. It has the ability to display numbers, characters and graphics. It has an inbuilt refreshing circuit, thereby relieving the CPU from the task of refreshing. LCD discussed has total of 14 pins.

**LCD pin Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Symbol** | **I/O** | **Description** |
| 1 | Vss | - | Ground |
| 2 | Vcc | - | +5V Power Supply |
| 3 | Vee | - | Power Supply to contrast |
| 4 | RS | I | RS = 0 to select command register |
| 5 | R/W | I | RS = 1 to select data register |
| 6 | EN | I/O | Enable |
| 7 to 14 | D0 to D8 | I/O | 8 bit data bus |

**Table 3.4 LCD Pin description**

In recent years the LCD is finding widespread use replacing LEDs this is due to following reasons:

1. The declining prices of LCDs
2. The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and few characters.
3. Incorporation of a refreshing controller in to LCD, there by relieving the CPU of the task of refreshing the LCD. In contrast LCD must be refreshed by CPU to keep displaying the data.

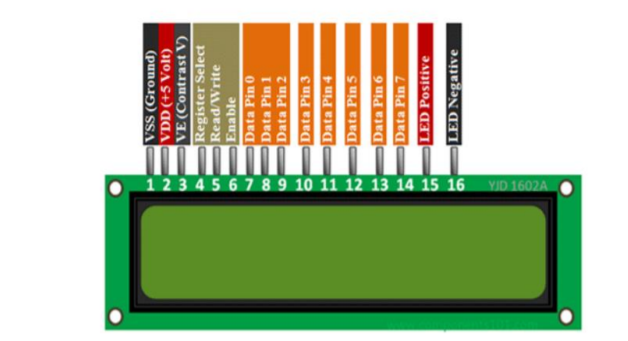
**LCD Command Codes**

|  |  |
| --- | --- |
| **Code(HEX)** | **Command to LCD Instruction Register** |
| 1 | Clear display screen |
| 2 | Return home |
| 4 | Decrement cursor (shift cursor to left) |
| 6 | Increment cursor (shift cursor to right) |
| 80 | Force cursor to the beginning of first line |
| C0 | Force cursor to the beginning of second line |
| 38 | 2 lines and 5x7 matrix |

**Table 3.5 LCD command codes**

|  |  |
| --- | --- |
| Terminal 1 | GND |
| Terminal 2 | +5V |
| Terminal 3 | Mid terminal of potentiometer(for brightness control) |
| Terminal 4 | Register select(RS) |
| Terminal 5 | Read/Write (RW) |
| Terminal 6 | Enable(EN) |
| Terminal 7 | DB0 |
| Terminal 8 | DB1 |
| Terminal 9 | DB2 |
| Terminal 10 | DB3 |
| Terminal 11 | DB4 |
| Terminal 12 | DB5 |
| Terminal 13 | DB6 |
| Terminal 14 | DB7 |
| Terminal 15 | +4.2-5V |
| Terminal 16 | GND |

**Table 3.6 LCD Specifications**



**Fig 3.18 LCD Display**

**3.5 SOFTWARE REQUIREMENTS:**

* + - 1. **ARDUINO IDE:**

****

**Fig 3.19 Arduino IDE**

* The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The middle section of the IDE is a simple text editor that where we can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.
* Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and the ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.
* Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package.
* 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 toolchain, also included with the IDE distribution.

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 toolchain, also included with the IDE distribution.

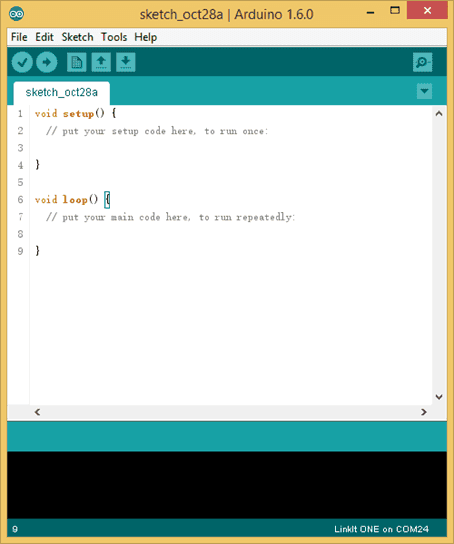
A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions:

**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.

**Arduino - Installation**

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.



**Fig 3.20 Arduino Installation**

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

**Step 1** − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.



**Fig 3.21 USB cable**

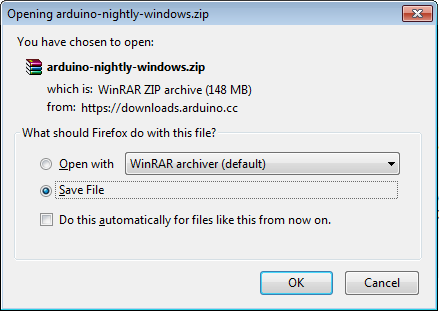
In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

**Step 2 − Download Arduino IDE Software.**

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



**Fig 3.22 A to Mini-B cable**



**Fig 3.23 Arduino opening**

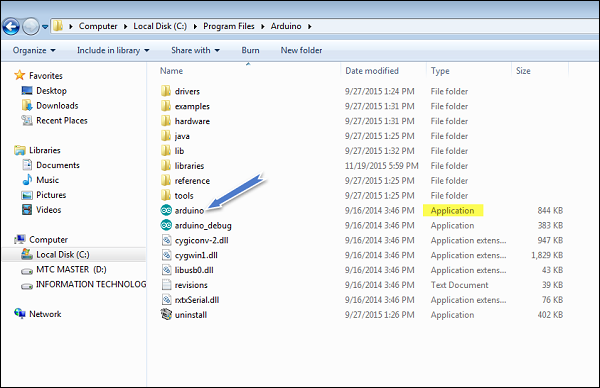
**Step 3 − Power up your board.**

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

**Step 4 − Launch Arduino IDE.**

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.



**Fig 3.24 Arduino Launch**

**Step 5 − Open your first project.**

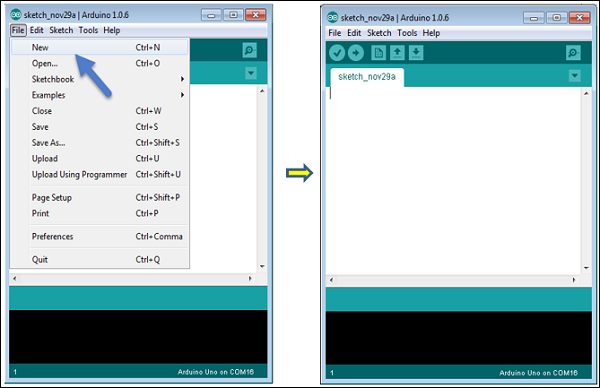
Once the software starts, you have two options −

* Create a new project.
* Open an existing project example.

To create a new project, select File→ New.

To open an existing project example, select File → Example → Basics → Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.



**Fig 3.25 Create new project**

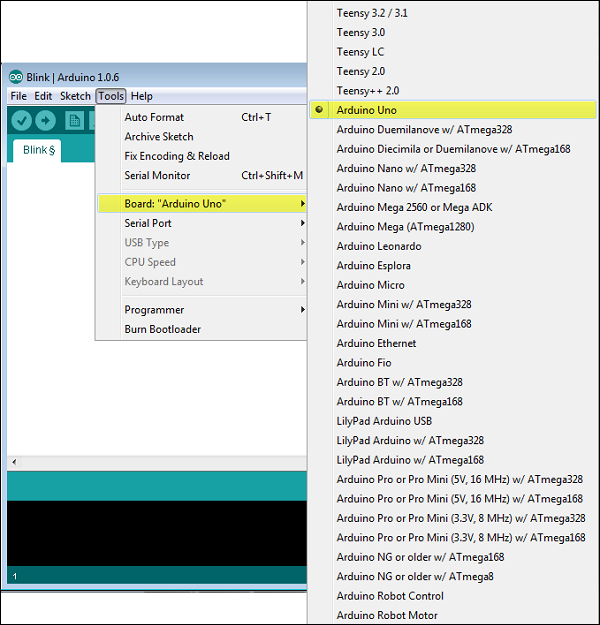
Open Project

**Fig 3.26 Blink**

**Step 6 − Select your Arduino board.**

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

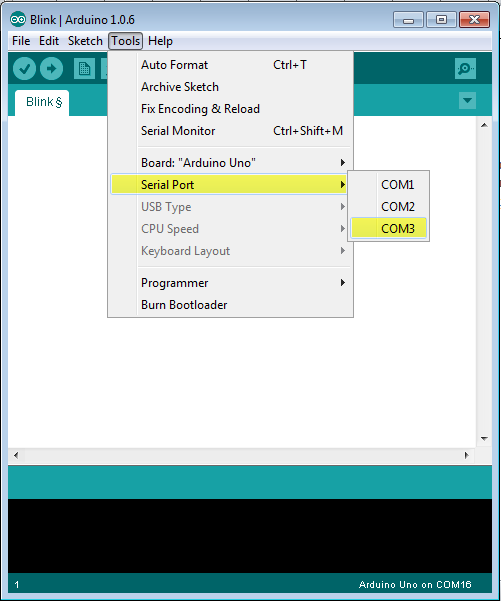


**Fig 3.27 Name Board selection**

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

**Step 7 − Select your serial port.**

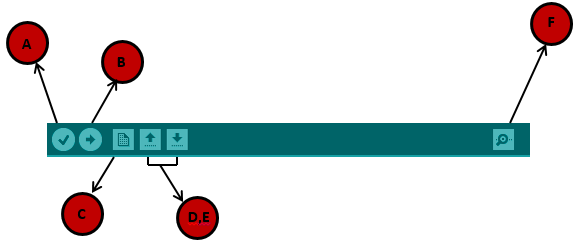
Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



**Fig 3.28 Port selection**

**Step 8 − Upload the program to your board.**

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



**Fig 3.29 Task Bar**

**A** − Used to check if there is any compilation error.

**B** − Used to upload a program to the Arduino board.

**C** − Shortcut used to create a new sketch.

**D** − Used to directly open one of the example sketch.

**E** − Used to save your sketch.

**F** − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software

1. **Embedded C:**

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

### What is an Embedded System?

An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

Embedded Systems can not only be stand-alone devices like Washing Machines but also be a part of a much larger system. An example for this is a Car. A modern day Car has several individual embedded systems that perform their specific tasks with the aim of making a smooth and safe journey.

Some of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, etc.

### Programming Embedded Systems

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

### Factors for Selecting the Programming Language

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

* **Size:** The memory that the program occupies is very important as Embedded Processors like Microcontrollers have a very limited amount of ROM.
* **Speed:** The programs must be very fast i.e. they must run as fast as possible. The hardware should not be slowed down due to a slow running software.
* **Portability:** The same program can be compiled for different processors.
* **Ease of Implementation**
* **Ease of Maintenance**
* **Readability**

Earlier Embedded Systems were developed mainly using Assembly Language. Even though Assembly Language is closest to the actual machine code instructions, the lack of portability and high amount of resources spent on developing the code, made the Assembly Language difficult to work with.

There are other high-level programming languages that offered the above mentioned features but none were close to C Programming Language.

### Introduction to Embedded C Programming Language

Before going in to the details of Embedded C Programming Language and basics of Embedded C Program, we will first talk about the C Programming Language.

The C Programming Language, developed by Dennis Ritchie in the late 60’s and early 70’s, is the most popular and widely used programming language. The C Programming Language provided low level memory access using an uncomplicated compiler (a software that converts programs to machine code) and achieved efficient mapping to machine instructions.

The C Programming Language became so popular that it is used in a wide range of applications ranging from Embedded Systems to Super Computers.

Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

### Difference between C and Embedded C

There is actually not much difference between C and Embedded C apart from few extensions and the operating environment. Both C and Embedded C are ISO Standards that have almost same syntax, datatypes, functions, etc.

Embedded C is basically an extension to the Standard C Programming Language with additional features like Addressing I/O, multiple memory addressing and fixed-point arithmetic, etc.

C Programming Language is generally used for developing desktop applications whereas Embedded C is used in the development of Microcontroller based applications.

### Basics of Embedded C Program

Now that we have seen a little bit about Embedded Systems and Programming Languages, we will dive in to the basics of Embedded C Program. We will start with two of the basic features of the Embedded C Program: Keywords and Datatypes.

#### Keywords in Embedded C

A Keyword is a special word with a special meaning to the compiler (a C Compiler for example, is a software that is used to convert program written in C to Machine Code). For example, if we take the Keil’s Cx51 Compiler (a popular C Compiler for 8051 based Microcontrollers) the following are some of the keywords:

* bit
* sbit
* sfr
* small
* large

These are few of the many keywords associated with the Cx51 C Compiler along with the standard C Keywords.

#### Data Types in Embedded C

Data Types in C Programming Language (or any programming language for that matter) help us declaring variables in the program. There are many data types in C Programming Language like signed int, unsigned int, signed char, unsigned char, float, double, etc. In addition to these there few more data types in Embedded C.

The following are the extra data types in Embedded C associated with the Keil’s Cx51 Compiler.

* bit
* sbit
* sfr
* sfr16

The following table shows some of the data types in Cx51 Compiler along with their ranges..

|  |  |  |
| --- | --- | --- |
| ***Data Type*** | ***Bits (Bytes)*** | ***Range*** |
| bit | 1 | 0 or 1 (bit addressable part of RAM) |
| signed int | 16 (2) | -32768 to +32767 |
| unsigned int | 16 (2) | 0 to 65535 |
| signed char | 8 (1) | -128 to +127 |
| unsigned | 8 (1) | 0 to 255 |
| float | 32 (4) | ±1.175494E-38 to ±3.402823E+38 |
| double | 32 (4) | ±1.175494E-38 to ±3.402823E+38 |
| sbit | 1 | 0 or 1 (bit addressable part of RAM) |
| sfr | 8 (1) | RAM Addresses (80h to FFh) |
| sfr16 | 16 (2) | 0 to 65535 |

**Fig 3.30 Cx51 Compiler along with their ranges**

### Basic Structure of an Embedded C Program (Template for Embedded C Program)

The next thing to understand in the Basics of Embedded C Program is the basic structure or Template of Embedded C Program. This will help us in understanding how an Embedded C Program is written.

The following part shows the basic structure of an Embedded C Program.

* + Multiline Comments . . . . . Denoted using /\*……\*/
  + Single Line Comments . . . . . Denoted using //
  + Preprocessor Directives . . . . . #include<…> or #define
  + Global Variables . . . . . Accessible anywhere in the program
  + Function Declarations . . . . . Declaring Function
  + Main Function . . . . . Main Function, execution begins here  
    {  
    Local Variables . . . . . Variables confined to main function  
    Function Calls . . . . . Calling other Functions  
    Infinite Loop . . . . . Like while(1) or for(;;)  
    Statements . . . . .  
    ….  
    ….  
    }
  + Function Definitions . . . . . Defining the Functions  
    {  
    Local Variables . . . . . Local Variables confined to this Function  
    Statements . . . . .  
    ….  
    ….  
    }

#### Different Components of an Embedded C Program

**Comments:** Comments are readable text that are written to help us (the reader) understand the code easily. They are ignored by the compiler and do not take up any memory in the final code (after compilation).

There are two ways you can write comments: one is the single line comments denoted by // and the other is multiline comments denoted by /\*….\*/.

**Pre-processor Directive**

* A pre-processor Directive in Embedded C is an indication to the compiler that it must look in to this file for symbols that are not defined in the program.
* In C Programming Language (also in Embedded C), Preprocessor Directives are usually represented using #include… or #define….
* In Embedded C Programming, we usually use the preprocessor directive to indicate a header file specific to the microcontroller, which contains all the SFRs and the bits in those SFRs.
* In case of 8051, Keil Compiler has the file “reg51.h”, which must be written at the beginning of every Embedded C Program.

**Global Variables:** Global Variables, as the name suggests, are Global to the program i.e. they can be accessed anywhere in the program.

**Local Variables:** Local Variables, in contrast to Global Variables, are confined to their respective function.

**Main Function:** Every C or Embedded C Program has one main function, from where the execution of the program begins.

* When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.
* The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.
* Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human’s thought process more closely than does the equivalent assembly code. [25]The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.
* By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C.
* All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware if need be, yet provides you the functionality and abstraction of a higher order language.

1. **Telegram Bots**

**Bots: An introduction for developers**

Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to our Bot API.

**What can I do with bots?**

A chat with @TechCrunchBot also showing search results from the @gif inline-bot

* To name just a few things, you could use bots to:
* Get customized notifications and news. A bot can act as a smart newspaper, sending you relevant content as soon as it's published.
* Integrate with other services. A bot can enrich Telegram chats with content from external services.
* Gmail Bot, GIF bot, IMDB bot, Wiki bot, Music bot, Youtube bot, GitHub bot
* Accept payments from Telegram users. A bot can offer paid services or work as a virtual storefront.
* Demo Shop Bot
* Create custom tools. A bot may provide you with alerts, weather forecasts, translations, formatting or other services.
* Markdown bot, Sticker bot, Vote bot, Like bot
* Build single- and multiplayer games. A bot can offer rich HTML5 experiences, from simple arcades and puzzles to 3D-shooters and real-time strategy games.
* GameBot, Gamee
* Build social services. A bot could connect people looking for conversation partners based on common interests or proximity.
* Do virtually anything else. Except for dishes — bots are terrible at doing the dishes.

**How do bots work?**

* At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:
* Send messages and commands to bots by opening a chat with them or by adding them to groups. This is useful for chat bots or news bots like the official TechCrunch bot.
* Send requests directly from the input field by typing the bot's @username and a query. This allows sending content from inline bots directly into any chat, group or channel.
* Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the Telegram API. We call that interface our Bot API.

**How do I create a bot?**

There's a… bot for that. Just talk to BotFather (described below) and follow a few simple steps. Once you've created a bot and received your authorization token, head down to the Bot API manual to see what you can teach your bot to do.



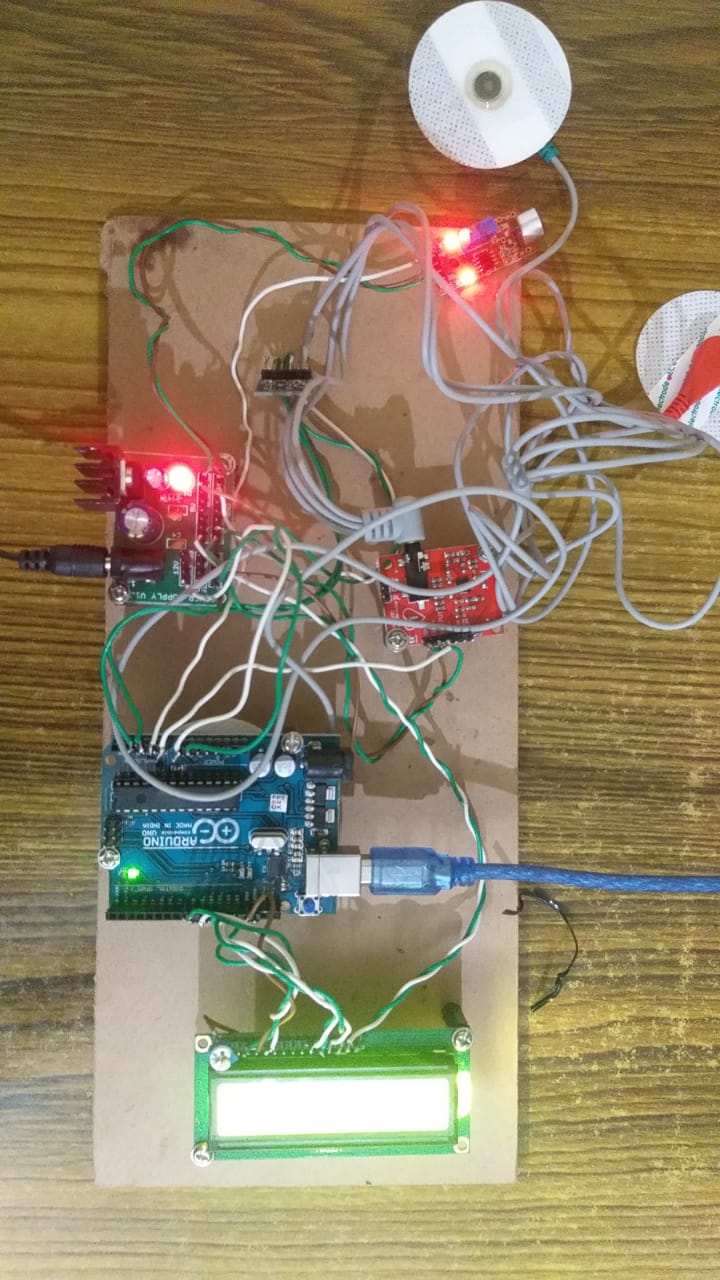
**Fig 3.31 The Botfather**

**CHAPTER 4**

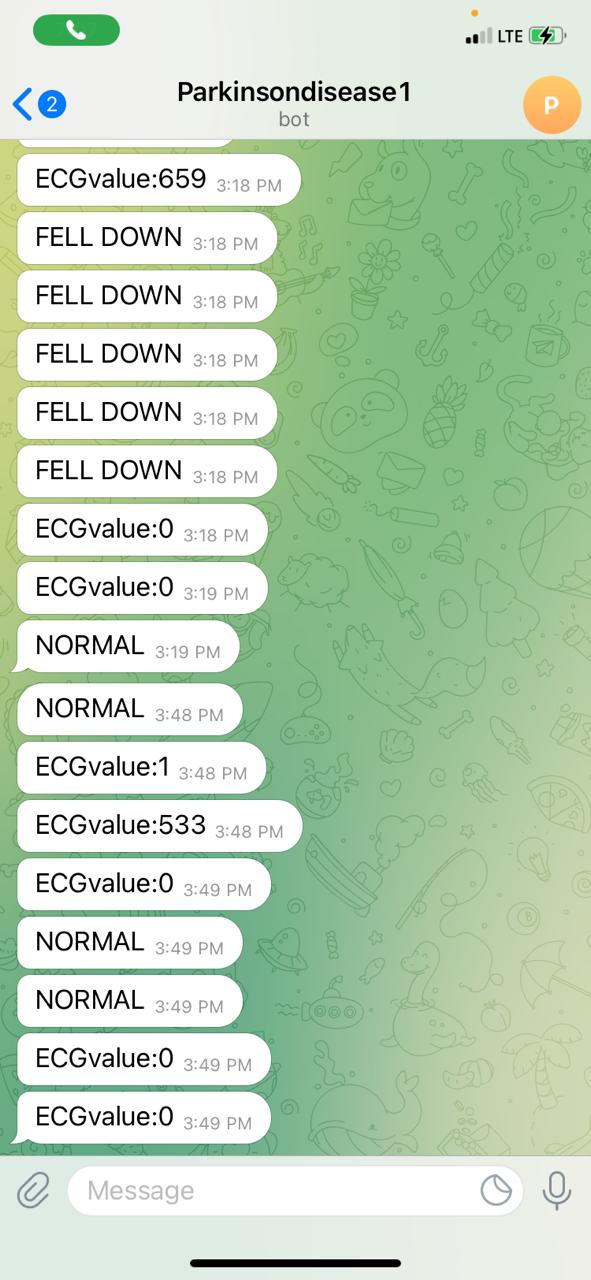
**RESULTS AND DISCUSSIONS**

**4.1 EXPERIMENTAL RESULT**

All the sensors mentioned above were implemented and the sensors worked as expected. Three people were chosen to test the developed prototype to ensure that it worked properly. When tested on healthy patients, the prototype demonstrated promising results. Although we were unable to test the prototype on Parkinson's disease patients, a random individual was chosen to replicate the behaviour of a Parkinson's disease patient, and the prototype was tested on that person, yielding a positive result.



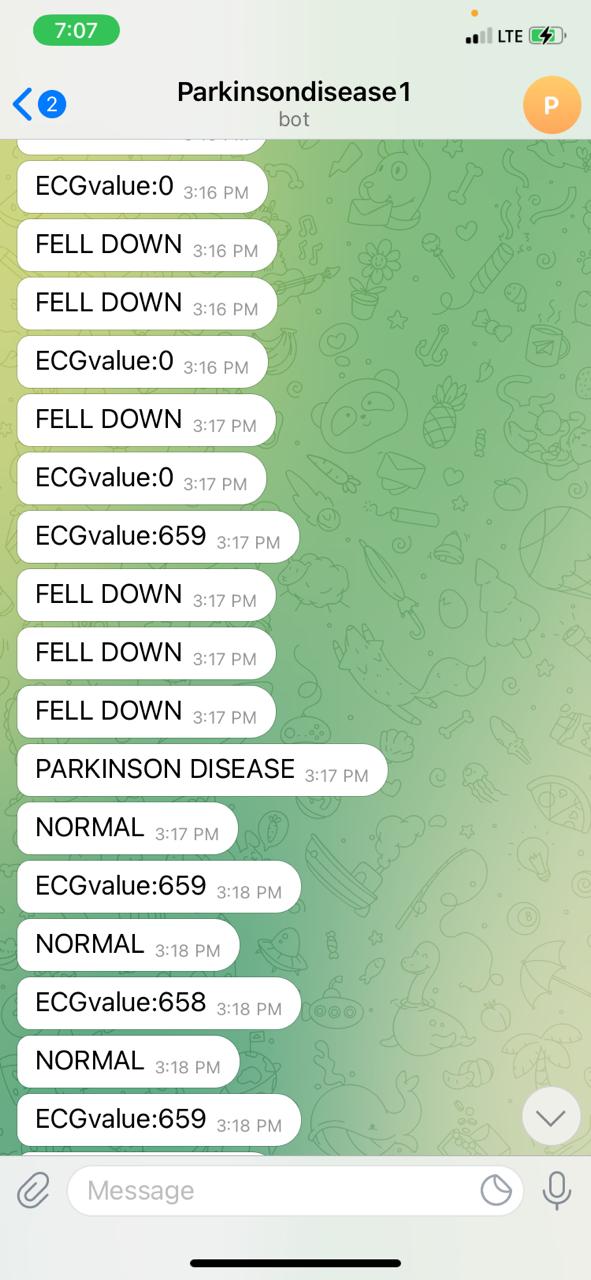
**Fig 4.1 Results**



**Fig 4.2 Values sent to the caretakers mobile on Telegram**

The values which are calculated using the sensors in the prototype is being analysed and displayed on the LCD module in the apparatus. As we have connected a Wi-Fi module, we can intimate all the values read on the sensors to the caretakers device using the Telegram Bot.

It displays all the sensors values with respect to its threshold value, intimates the caretaker as soon as the patient falls off, and many other characters of the device implemented.



**Fig 4.3 Values sent to the caretakers mobile on Telegram**

**CONCLUSION**

The developed system could easily be used in medical offices due to its portability, low cost, low power and fast response in terms of output availability. It could therefore help doctors in fast diagnoses of early-stage Parkinson’s patients, thus enhancing the probabilities to help them to live better with more effective medical treatments. The current status of the system proves its capability to follow movements and tremors that are typically involved in Parkinson’s disease. Automated Emergency Situation Detection. Monitoring Person conditions. Intimation Alert to Care Takers.

This work presents a method to acquire and analyse the various symptoms from Parkinson’s disease (PD) patients. This prototype has an accelerometer, sound sensor, gyroscope, EMG sensor and Arduino Uno to acquire the tremor signal, speech fluctuations and muscle movement. The Arduino was connected to the computer using a USB cable. Both the accelerometer and Arduino Uno were mounted on the glove to get the signal from fingertip, wrist, and forearm. The sound sensor was placed near the mouth of the patient to record their voice. The EMG Sensor was used to measure the small electrical signals produced by our muscles as we move them, such as raising our arm or clenching our fist. The prototype presents a basic medical wearable device that could be used to study the symptoms from the PD patients and assist them with the right treatment.

**FUTURE SCOPE**

We believe we have room for improvement and that would be to design a more sensitive and one-time-calibrated device with minimum intervention by medical specialists. Our model is a very specific approach for tackling the problem. We can have a more generalized approach to tackle the problem addressed, to name a few we can have self-stabilizing gloves where one can perform all the action with his/her hands. We can also have a self-stabilizing pen, vest, or walking stick. This will be a ray of hope for people suffering from Parkinson’s and other similar diseases. All of these use the same technology, with an extension from the design.

**REFERENCES**

[1] C.M. Tanner and S.M. Goldman, “EPIDEMIOLOGY OF PARKINSON’S DISEASE,” Neurologic Clinics, vol. 14, no. 2, pp. 317-335, 1996.

[2] J.J. Pasternak and W.L. Lanier, “Diseases Affecting the Brain: Parkinson's Disease,” in Stoelting's Anesthesia and Co-Existing Disease, Elsevier, Inc, 2018, pp. 265-303.

[3] L.V. Kalia and A.E. Lang, “Parkinson’s disease,” The Lancet, vol. 386, pp. 896-912, 2015.

[4] J.A. Stamford, P.N. Schmidt and K.E. Friedl, “What Engineering Technology Could Do for Quality of Life in Parkinson’s Disease: A Review of Current Needs and Opportunities” IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 6, pp. 1862-1872, 2015.

[5] C. Fazekas, T. Vörös, Z. Keresztényi, G. Kozmann, J Laczkó, “Computer aided interactive remote diagnosis of Parkinsonians” Stud Health Technol Inform, vol. 90, pp. 572-576, 2002.

[6] D. Giansanti, V. Macellari and G. Maccioni, “Telemonitoring and Telerehabilitation of Patients with Parkinson’s Disease: Health Technology Assessment of a Novel Wearable Step Counter” Telemedicine and e-Health, vol. 14, no. 1, pp. 76-83, 2008.

[7] A. Bermeo, Student Member, IEEE, M. Bravo, Student Member, IEEE, M. Huerta, Senior Member, IEEE and A. Soto, Member, IEEE, “A System to Monitor Tremors in Patients with Parkinson's Disease”, Latin American Congress on Biomedical Engineering CLAIB 2014, Buenos Aires, Argentina 29, October 2014, pp. 785-788.

[8] Lemoyne, T. Mastroianni, M. Cozza, C. Coroian, and W. Grundfest, “Implementation of an iPhone for characterizing Parkinson’s disease tremor through a wireless accelerometer application,” Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Int. Conf., vol. 2010, pp. 4954–4958, 2010, doi: 10.1109/iembs.2010.5627240

[9] L. Cunningham, C. Nugent, G. Moore, D. Finlay, and D. Craig. “Identifying Fine Movement Difficulties in Parkinson's Disease Using a Computer Assessment Tool.”, IEEE, 2020.

[10] H. Zhang, X. Chen, W.-Y. Lin, W.-C. Chou, and M.-Y. Lee, “A novel accelerometer-based method for the real-time assessment of Parkinson’s tremor,” in 2014 IEEE International Conference on Communication Problem-solving, 2014, pp. 87–90, doi: 10.1109/ICCPS.2014.7062224.

[11] A. Zhang et al., “Automated Tremor Detection in Parkinson’s Disease Using Accelerometer Signals,” in 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2018, pp. 13–14, doi: 10.1145/3278576.3278582.

[12] S. Zabberoni, G. Carlesimo, A. Peppe, C. Caltagirone, and A. Costa, “Does dopamine depletion trigger a spreader lexical-semantic activation in Parkinson’s disease? evidence from a study based on word fluency tasks,” Parkinson’s Disease, vol. 2017, 2017.

[13] B. Gupta and A. Bala, "Parkinson's disease in India: An analysis of publications output during 2002-2011," International Journal of Nutrition, Pharmacology, Neurological Diseases, vol. 3, pp. 254-262, 2013.

[14] J. Jankovic, "Parkinson’s disease: clinical features and diagnosis," Journal of Neurology, Neurosurgery &amp; Psychiatry, vol. 79, p. 368, 2008. M. Nyemi, “Vehicle safe drive system. kakamega,” 2012.

[15] J. McNames, P. M. R. Aboy, and A. Greenberg, "Method and apparatus for continuous measurement of motor symptoms in Parkinson’s disease and essential tremor with wearable sensors," United States Patent, 2009.