2012

Brain Controlled Switch

Dimple Bhuta
Virginia Commonwealth University

BRAIN CONTROLLED SWITCH

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

DIMPLE BHUTA
Bachelor of Engineering (Electronics), Mumbai University, India, 2009

Director: DING-YU FEI, PH.D.
ASSOCIATE PROFESSOR, DEPARTMENT OF BIOMEDICAL ENGINEERING

Virginia Commonwealth University
Richmond, Virginia
August, 2009

ACKNOWLEDGEMENTS

I am extremely grateful to my advisor Dr. Ding-Yu Fei and my co-advisor Dr. Ou Bai for giving me an opportunity to work on this project. This thesis wouldn't have been possible without their continuous guidance, motivation and support at every step of my research. Throughout my project they have encouraged me to develop independent thinking and research skills. I would also like to thank my thesis committee member Dr. Azhar Rafiq. Thank you for your insightful comments and all your help in the completion of my project. I would like to thank Ms. Cai-Ting Fu for her knowledge and suggestions during my research.

I would really like to thank my lab mates and friends Deepak Kumbhare, Vaishnavi Karnad, Dandan Huang and Sayak Bhattacharya for making my graduate study such an enjoyable experience.

Last but not the least, I am really grateful to my parents Dr. Leena and Dr. Chitranjan Bhuta who always believed in me, supported me unconditionally and showed me that there is a great joy in the intellectual pursuit.

Table of Contents

LIST OF ABBREVIATIONS

BCS	Brain Controlled Switch device
BCI	Brain Computer Interface
EEG	Electroencephalography
SSVEP	Steady-state visual evoked potentials
ALS	Amyotrophic lateral sclerosis
FPR	False positive rate
TPR	True positive rate
ERS	Event-related synchronization
ERD	Event-related desynchronization
EMI	Electromagnetic interference
RF	Radio frequency interference
EMG	Electromyography
ECG	Electrocardiography
IA	Instrumentation Amplifier
UART	Universal Asynchronous receiver transmitter
ADC	Analog to Digital Converter
LED	Light Emitting Diode
ISR	Interrupt Service Routine
IPCx	Interrupt Priority Control Bits

SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
IPEx	Interrupt Enable Bit
IFSx	Interrupt Flag Status Bits
F_{OSC}	System Clock Frequency
T_{OSC}	System Clock Period
F_{cy}	Instruction Cycle Frequency
F_{Cy}	Instruction Cycle Period
MIPS	Million Instructions Per Second
BPS	Bits Per Minute

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABSTRACT

Brain Controlled Switch

By Dimple Bhuta.

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University, 2009

Major Director: Dr. Ding-Yu Fei

Associate Professor, Department of Biomedical Engineering

This study aims at designing and implementing a single channel stand-alone Brain-Controlled Switch (BCS) device, which records the electroencephalography (EEG) signals from the scalp using electrodes, amplifies it, eliminates interferences (associated with the EEG signals) and processes the EEG signals to extract and decode temporal signal features to determine user's intention of regulating an external switch.

The design of our "brain-controlled switch" device is implemented using a bio-potential amplifier and a microcontroller. The bio-potential amplifier amplifies the EEG signals to a level sufficient for processing, eliminates interferences and ensures patient safety. The microcontroller (dsPIC30F4013) digitizes the amplified and conditioned analog EEG signals from the bio-potential amplifier, extracts the desired signal features for decoding and prediction of user's intention and accordingly operates the external switch.

When the user concentrates on an external visual stimulus or performs externally triggered movement (hand movement or motor imagery movement), a reproducible pattern appears in user's EEG frequency bands. The analysis of these patterns is used to decode and predict user's intention to operate an external switch. To realize our "brain-controlled switch", we explored two EEG sources: steady-state visually evoked potentials (SSVEP) and beta rebounds, which are patterns generated in the EEG frequency bands associated with focusing on an external visual stimulus or performing externally triggered movements.

In case of SSVEP based brain controlled switch, a repetitive visual stimulus (LED flickering at a specified frequency) was used. When the user concentrates on the flickering LED, a dominant fundamental frequency (equivalent to the flickering frequency) appears in the spectral representation of the EEG signals recorded at occipital lobes. Our microcontroller implemented a digital band pass filter to extract the frequency band containing this fundamental frequency and continuously took an average of the amplitude power every predetermined time interval. Whenever the amplitude average power exceeded the preset power threshold the external switch was turned ON. A healthy subject participated in this study, and it took approximately 3.14 ± 1.81 seconds of active concentration for the subject to turn ON the switch in real time with a false positive rate of 1.17%.

In case of beta rebound based brain controlled switch, the subject was instructed to perform a brisk hand movement following an external synchronization signal. Our design focused on the post-movement beta rebound which occurs after the cessation of the movement to operate the external switch. Our microcontroller in this case implemented a digital band pass filter to extract the beta band and continuously took an average of its amplitude power every predetermined time interval. Whenever the amplitude average power exceeded the preset power threshold the

external switch was turned ON. It took approximately 12.23 ± 7.39 seconds of active urging time by the subject to turn ON the switch in real time with a false positive rate of 9.33%.

Thus we have designed a novel stand-alone BCS device which operates an external switch by decoding and predicting user's intentions.

CHAPTER 1

INTRODUCTION

1.1 Background

Since the discovery of electroencephalography (EEG) by Hans Berger in 1929 [1], people have speculated that it could be used as muscle-independent communication and control channel (which doesn't use brain's regular pathways of peripheral nerves and muscles) between brain and external devices such as computers, wheelchairs or robotic arms. Hence it is termed as a brain-computer interface (BCI). The target population of the BCI devices is "totally locked-in" patients: their mind is conscious and alert however they lose their ability to move and communicate due to the loss of their motor function. Hemorrhage of brain stem, stroke and tumors, encephalitis, and brain injuries localized in the ventral midbrain causes locked-in syndrome [2]. Also neuromuscular diseases such amyotrophic lateral sclerosis (ALS caused by progressive degeneration of central and peripheral motor neurons), multiple sclerosis and cerebral palsy causes total-locked in syndrome. In "total-locked in" condition, only 10% of the population live more than 10 years following diagnosis and more than 90% of the patients do not choose to prolong life by accepting ventilation due to the anticipated loss of the ability to communicate[3, 4]. In classic "locked-in" syndrome vertical eye movement as well as eye blinks remains intact, which could be used as a basic means of communication to answer questions, give simple commands, or even operate a word processing program. However in total "locked-in syndrome" even the eye muscles are paralyzed, hence an EEG based BCI device can provide basic communication capabilities to these patients with no muscle control, so that they could

express their wishes to caregivers or operate word processing programs and even control a robotic arm [5].

A BCI system mainly consists of four parts: signal recording, processing, predication of the user's intention and controlling the external device. Current BCI researchers record signals for the BCIs from the scalp or utilize signals recorded from inside the brain using implanted electrodes, thus based on the recording site BCI can be divided into non-invasive or invasive BCIs respectively [6]. The signals recorded from non-invasive BCIs generally passes through a bio-potential amplifier which has high amplification ability, high common mode rejection ratio CMRR and circuit protection features; to obtain an appropriate EEG signal for processing. In the signal processing part, signal features in terms of amplitude and/or frequency power are extracted. The prediction part determines user's intention by analyzing the extracted signal features that the user encodes if he/she wants to perform a previously specified task. The extracted signal features are translated into real time commands to operate and control the external device.

Based on the site of acquiring brain signals, the BCI systems can be divided into two types:

(a) Invasive BCI: It provides neuronal signals of best quality as they are recorded from electrodes placed inside the brain. Brain signals can be recorded from small samples of neurons in the single cortical areas, local field potentials (LPF) or large neurons from multiple cortical areas[7]. The signals obtained from the invasive BCI's are very specific to the movement (can produce position and velocity information of the movement), achieve high speed of communication and can be used to control devices with multiple degrees of freedom. The action potentials (neuronal spikes from multiple recording sites) and the LPF (local field potentials) recorded from electrodes inserted into the cortical

tissues were used to operate robotic arms which requires multiple degrees of freedom [8-14]. However these methods are far from practical clinical applications as the electrodes can't be used for a long time, as implantation of the electrodes can further damage the brain tissue by central nervous system infection and also the conductivity of electrodes decreases over time as the body's immune response encapsulates the electrodes. Also most of these experiments have been carried out in non-human primates; hence there is a large gap in applying invasive BCI methods to the target population of "locked-in" patients.

(b) Non-Invasive BCI: It is commonly known as EEG-based BCI system. EEG refers to the mean brain electrical activity arising from billions of neurons recorded non-invasively from the scalp (generally motor cortex or somatosensory cortex) [15]. A composite EEG signal comprises of different rhythms (frequencies bands) produced by distinct cortical areas. They are delta (< 4Hz present frontally in adults and in the posterior lobe in children associated with deep sleep or anesthesia), theta (4-8Hz), alpha (8–12 Hz predominant over occipital, parietal and posterior temporal regions, and are associated with relaxed focus or readiness), beta (13–30 Hz predominant over frontal and central lobes, and are associated with active concentration and alert thinking) and gamma (above 30Hz dealing with mental representations). The EEG based BCI systems tries to decipher its user's intention and decisions through measurements of the combined electrical activity of massive neuronal populations. Hence the quality of the signal recorded from EEG-based BCI system is poor (compared to invasive systems, as the EEG signals are contaminated with neural sources of noise (such as EEG features not used for communication) as well as non-neural artifacts (such as power line noise). It also lacks

specificity (information about the position and velocity information of the movement). However compared to invasive methods, it is robust over time, inexpensive, convenient to use and has low risk in implantation. Several successful EEG based methods have been reported such P300 based letter selection system for 51yr old ALS patient with an accuracy of 83% over 2.5years[16] , mu rhythm based one-dimensional cursor control (up-down of the screen)[17, 18], self-regulated mu rhythm[19],slow cortical potentials based computer-aided spelling system[17, 18], motor imagery based brain controlled switch[20] and visual evoked potential (VEP)[21-24]. EEG based methods appear to be similar to conventional skills which no longer requires intense concentration once learned. Some BCI protocols ask that the user employs very specific motor imagery (e.g., imagery of right or left hand movement) or other mental tasks to produce the EEG features the system uses as control signals. Current EEG-based BCI systems have transfer rates between 5-25 bits/minutes [25, 26]. Although these transfer rates aren't sufficient for controlling neuro-prosthetic devices such as robotic arms, they try to offer some practical solutions (e.g. yes/no answer reply by switch control, one and two dimensional cursor control and wheel chair control) for "locked-in" patients in the near future.

1.2 Objective: The objective of this study is to design a real world single channel BCS device. The design is based on the assumption that our target population of "locked-in" patients can still think about moving and the brain areas corresponding to the movement generate patterns that represent the expected value of reward. Our goal is to record these movement intentions, interpret them and use them as control signals to operate a simple ON/OFF switch.

Our device consists of an analog front end circuit (or a bio-potential amplifier) and the digital part. The analog part records the EEG signals from scalp electrodes, amplifies them to a level sufficient for signal processing, reduces the non-neural EEG signal artifacts, filters them to remove the dc offset and perform anti-aliasing. The digital part of our system is designed to convert analog EEG signals into digital signals for processing, extract the features of our interest from the overall available EEG features, eliminate interferences and implement a prediction algorithm to operate an external switch.

To make our design compatible with the real world switch we have tried to implement a design which lets the user decide when to switch ON the external switch by performing a task. A couple of research groups have explored the self-paced or asynchronous EEG based BCI to differentiate between "Intentional Control" state (subject performs a task to indicate movement intention)" and "No Control state" (rest state)[27-29]. We have utilized the principle of "Intentional control" and "No control" state by providing system cues to realize an asynchronous BCS.

We consider the following properties in developing a real world brain-controlled switch:

(a) Self-paced: Incase of the real-world light switch, the switch is turned ON/OFF by the user only when he/she intends to do so by performing a movement task. Similarly in case of BCS, the users would pay attention to the system cues only when they want to turn the switch ON/OFF. Birch's group addressed that the users may perform a certain motor task only when they want to turn ON the switch also called as "Intentional Control state", whereas the users may carry out normal thinking (any task such as relaxing, day dreaming or thinking about a problem) when the users does not wish to operate the switch also called as the "No control" or the rest state[28]. In our experiment, the subject was asked to perform a brisk hand movement in synchronization with an external visual

go-signal as well as asked to concentrate on an external visual stimulus in the "Intentional Control" state. In the "NO control" states, the subject was asked to look away from the visual cue.

(b) Minimal false positive rate (FPR): In case of the real world light switch, the switch stays OFF in the "No Control" state; our BCS should also avoid false triggering, that is activation of the switch in the "NO control" state. Birch's group suggested that 0% FPR is required in practical applications and they achieved FPR below 1% in their offline or pseudo-online optimization studies [30, 31]. We have also used FPR to evaluate our switch's performance.

(c) Sensitivity: Real world switch allows its users multiple attempts until the switch is turned ON, since not all switches are sensitive enough that they can be turned ON in the single attempt. Also once activated the switch remains ON until the user intends to turn it OFF. This property is particularly important in case of a BCS as the EEG signal isn't strong enough and the EEG patterns associated with the goal signal (intension of movement) aren't consistent with time. In sensitivity measurement for switch's performance evaluation, switch performance is measured as the average of response time it takes to activate the switch in determined number of trials.

External stimulus based operation: BCI technology was proposed to establish a non-muscular communication and control channel between brain of "totally locked-in" patients and external devices. Two EEG based methods: beta-rebound and steady state visual potentials (SSVEP) were tested to extract the amplitude power feature of the EEG signal from a specific frequency band (beta band for event related potentials and narrow band including the frequency of the SSVEP signal).

(i) Beta rebound based method: Event-related potentials are electrical brain responses whose voltage amplitude increases (event-related synchronization or ERS) or decreases (event-related desynchronization or ERD) in a specific frequency band, in response to an externally triggered movement (hand movement or motor imagery task). These ERD/ERS patters are reproducible and are stable over time. In the beta band, following an ERD that occurs shortly before and during the movement, ERS with very high amplitude oscillations appear within 1 second interval post movement which is also called as beta rebound[32]. A neural network learns to recognize the ERS/ERD patterns associated with specific movements (e.g. right-hand or left hand) and, eventually, the patterns associated with simply thinking about these movements. After the neural network is trained, the system can recognize the EEG patterns associated with specific movements with impressive accuracy and can thereby control movement of a cursor or other external device[33]. In our experiment we have chosen to work with beta rebound patterns generated in-response to cessation of movement after the brisk movement. The beta band rebound is used by our algorithm to control the switch.

(ii) Steady State Visually Evoked Potentials (SSVEP) based method: They are "frequency coded signals" which can be recorded throughout the visual system. The fundamental frequency of the SSVEP is equal to the flashing or flickering frequency of the visual stimulus, thus the target the subject is gazing at can be identified by frequency analysis. The amplitude increase (associated with concentrating on visual stimuli) in the frequency band of the SSVEP can be used

to control brain-computer interfaces[34]. The frequency bandwidth in which the SSVEP signals are effectively observed is determined empirically to be approximately 6-24Hz[35]. The lower frequencies (6-8Hz) give high accuracy but slower speeds. The stimulus frequency in alpha wave range (8-12Hz) has high background alpha band EEG interference, thus decreasing the accuracy of amplitude measurement. However the beta band (13-28Hz) contains the fundamental frequency component without any interference and has the highest amplitude detection associated with corresponding flickering frequencies; hence the beta band frequency of 13Hz was selected as flickering LED frequency to control the SSVEP based BCS. SSVEP BCIs are considered to be dependent BCI as they depend on muscular control of gaze direction for their operation.

CHAPTER 2

METHODS

A number of portable, battery operated medical instruments are used in hospitals worldwide due to the recent advances in technology [36]. Also the rapidly expanding field of neuro-prosthetics aims at interfacing artificial devices with the brain [37]. The aim of my project is to develop a one channel portable EEG system to control the light switch. Hence it is termed as a brain-controlled switch. The circuit consists of two parts: an analog front-end circuit (bio-potential amplifier) and a digital circuit [38] show in Fig.1.

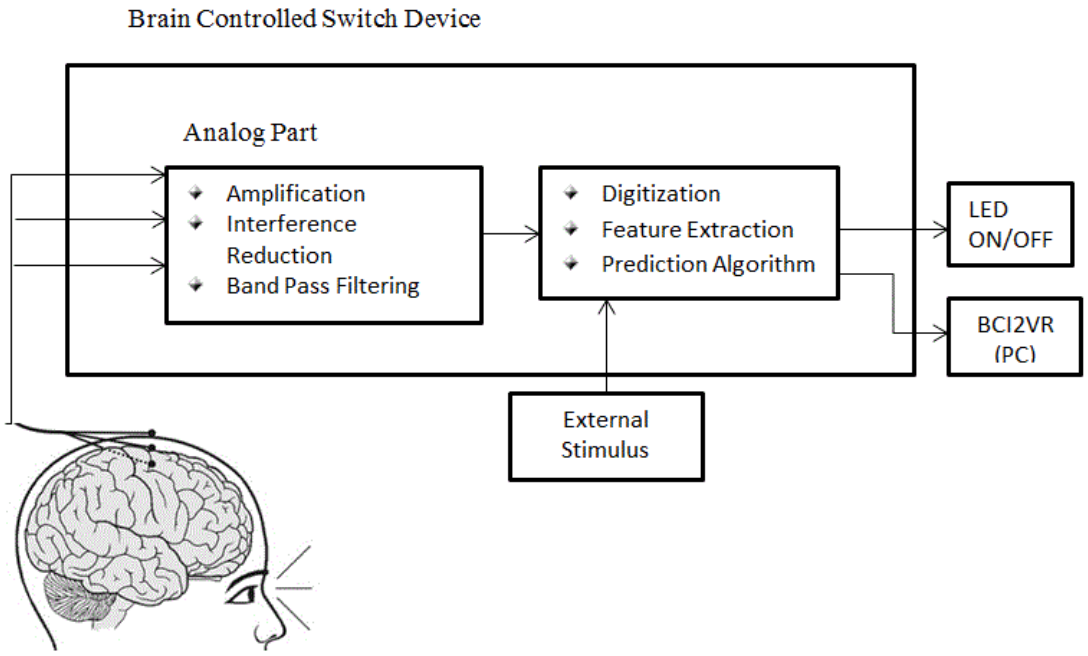


Fig.1. Block Diagram of the Brain Controlled Switch Device.

EEG signal amplitude is normally in the range of 0.5 to 100 microvolt's and it contains electrical and physiological artifacts hence it first passes through an analog part (also called the bio-

potential amplifier) which amplifies and conditions the EEG signal to a level sufficient for processing. The analog signal is than converted into digital samples by the microcontroller, features of interest from the EEG signal are extracted using digital filters and a prediction algorithm is implemented in the microcontroller based on external stimulus based operation performed to determine user's intent and accordingly operate the switch.

2.1 Hardware section:

A bio-potential amplifier is used in this section. The EEG signal recorded non-invasively from the scalp is the input to the one channel bio-potential amplifier. The EEG signal obtained from the electrodes is only few microvolts range and it is contaminated with electrical and physiological artifacts (or interferences). The electrical interference arises from electromagnetic interference (EMI), RF (radio frequency interference), cable movements, broken wire contacts, too much electrode gel or dried gel and power line interference (50/60Hz noise) [39, 40]. Physiological sources of interference are motion artifacts, electromyography (EMG) or muscle noise, eye motion or eye blinks, skin-electrode interference and sometimes even the electrocardiography (ECG) or heartbeats. EEG signal acquisition is very important for any BCI system [41]. Thus the bio-potential amplifiers must have high amplification ability (to amplify the microvolt EEG signals for further processing), ability to eliminate electrical interferences, have high input impedance and provide electrical protection to the circuit against high voltage inputs such as defibrillation shocks. To achieve these features, the designed bio-potential amplifier consists of pre-amplifier, main amplifier with filters, driven-right leg circuit (DRL), protection and isolation circuit shown in Fig. 2.

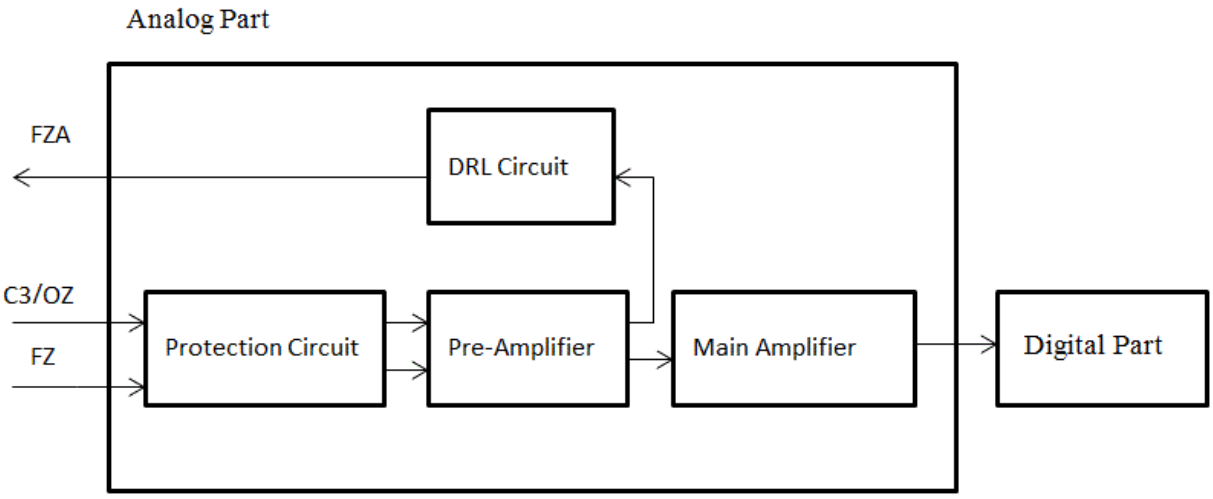


Fig.2. Block Diagram of the Analog Section.

The output signal from the bio-potential amplifier is the conditioned and amplified EEG signal is sent for digital signal processing to the dsPIC30F4013 microcontroller. The parts of the bio-potential amplifier are discussed in detail in this section and the schematic diagram is shown in appendix A.

2.1.1 Protection Circuit: Protection circuit protects against high input voltages, from electrostatic discharge, RF signals entering the system through electrode cables and accidental power supply contact. These large voltages can destroy the high impedance instrumentation amplifier, and the EEG circuitry[42].

As shown in the appendix A, a pair of differential signals enters the protection circuit through capacitors (capacitors filters out high frequency input interference and RF interference, if high frequency interference occurs the capacitor shorts) and resistors limit the current. The clamping diodes enables the inputs to the operational amplifier to be limited to the supply voltage plus the diode forward junction voltage (5.7-0.0 volts with reference to AGND or 3.1 (2.5(VGND) +0.7)

to -3.1 with reference to Virtual ground). When the voltages are above this level, diodes act as open circuits pulling harmful currents to ground potential thus protecting the EEG circuit and the user. The resistors after the diodes limit the current to the Instrumentation amplifier.

2.1.2 Pre-amplifier: An instrumentation amplifier (IA) AD620 is used as the pre-amplifier. It is a low cost, high accuracy instrument with excellent CMRR (AD620 datasheet).The IA is the heart of the bio- potential amplifier as it provides high input impedance, high CMRR and amplification ability which is needed for the accurate measurement of the EEG signals.

The high input impedance is required as it provides minimal loading to the signal being measured thus preventing signal distortion.

Pair of differential signals from the protection circuit is an input to the IA. The IA relates these signals. This relation is performed based on the assumption that both the signals will contain common mode signal (interferences of the EEG signal). When the differential signal is referenced to op-amp output zero voltage, the unknown common-mode signal found in the input signals is eliminated and the output is a pure differential signal. This property of the IA is called as the CMRR. High CMRR is essential for a good quality instrumentation amplifier[43].

IA also amplifies the pure differential voltages (EEG signal) by a factor of 10. This amplification factor (or gain of IA) is calculated as follows.

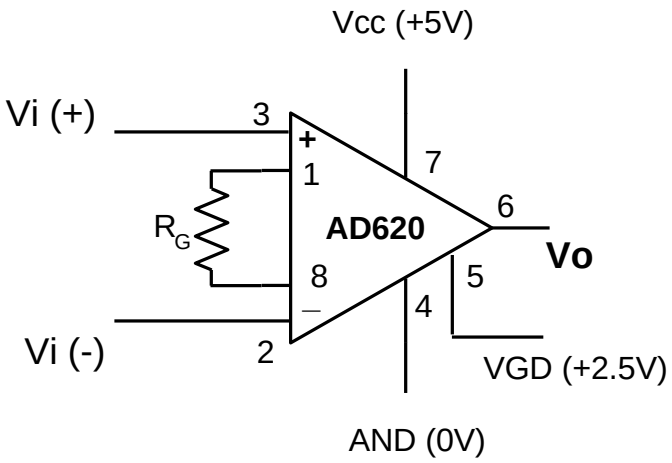


Fig.3. Circuit Diagram of the Instrumentation Amplifier.

Gain of the instrumentation amplifier is determined by R_G and is given by

$$\text{Gain } (G_{IA}) = \frac{49.4K\Omega}{R_G} + 1 ,$$

We selected R_G = 2*(2.7k) = 5.4 kΩ and therefore

$$G_{IA} = \frac{49.4K\Omega}{5.4K} + 1,$$

G_{IA} = 10.1

AD620 also provides low power operation which is beneficial while using multiple independent EEG channels.

2.1.3 Main amplifier: It provides the final amplification for the EEG signal and further helps to remove interferences by using a second order band pass filter. We are using two stages of first order band pass filter (passive high pass and active low pass) to realize a second order band pass filter (range 0.2-102.5 Hz), the design of which is shown in schematic A.

A passive high pass filter is used to remove DC voltage offsets and reduce the baseline drift[44]. It is a very essential part of the bio-amplifier as electrodes (made of gold, tin and silver) are polarizable accumulating electric charge on the surface of the electrode hence building up a large DC voltage offset which could saturate the amplifier (limit the voltage value up to 2.5V which doesn't contain any EEG). A second order active low pass filter is used to prevent aliasing that would otherwise occur when the signal is converted to digital samples by the ADC in the digital part. For example according to the sampling theorem our sampling frequency f_s must be greater than twice of the highest frequency; i.e. $f_s \geq (2 \times f_{max}) \geq (2 \times 102.5) \geq 205$ Hz. It also attenuates the muscle signal interference (EMG noise (1-5000) range) and the remaining RF interference.

The calculations of cut-off frequencies of the band pass filter and total gain of the pre-amplifier is shown below:

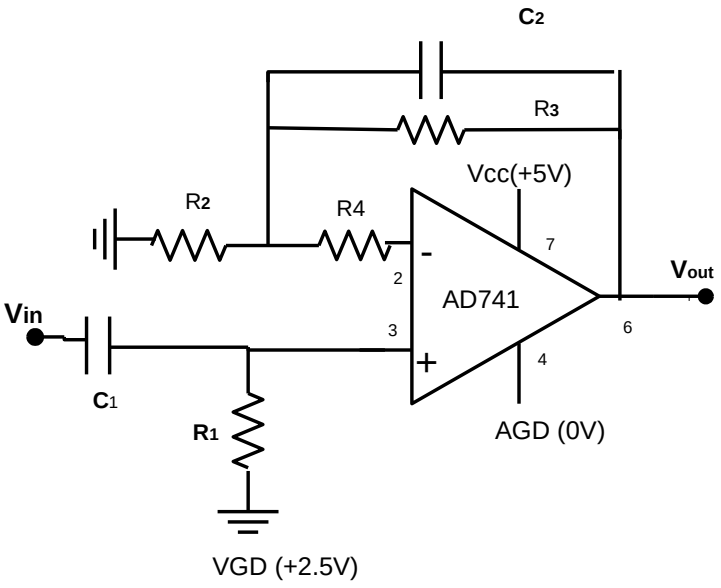


Fig.4. Circuit Diagram of the First-order Band Pass Filter (Active low pass and passive high pass).

Passive high pass filter cut-off frequency: $f_{c1} = \frac{1}{2\pi \cdot R_1 \cdot C_1}$

Active low pass filter cut-off frequency: $f_{c2} = \frac{1}{2\pi \cdot R_3 \cdot C_2}$

Gain of the Band pass filter: $G = \frac{V_0}{V_i} = \frac{R_3}{R_2} + 1$

(i) For the first stage of the first order band pass filter:

We have selected R_1 = 1MΩ, R_2 = 1KΩ+5KΩ (pot), R_3 = 100KΩ, C_1 = 1 μF, C2= 0.01uF

$f_{c11} = \frac{1}{2\pi \cdot R_1 \cdot C_1} = \frac{1}{2\pi \cdot 1M\Omega \cdot 1\mu F} = 0.159 = 0.2$ Hz, where we have selected f_{c11} as the cut-off frequency for the first stage high pass filter

$f_{c12} = \frac{1}{2\pi \cdot R_3 \cdot C_2} = \frac{1}{2\pi \cdot 100K\Omega \cdot 0.01\mu F} = 159.2$ Hz, where we have selected f_{c12} as the cut-off frequency for the first stage low pass filter

$G_1 = \frac{V_0}{V_i} = \frac{R_3}{R_2} + 1$, where we have selected

G_1 as the gain of the first stage amplifier and R_2 is a variable register (5KΩ pot and 1K register), we have selected $R_{21} = 1k$ and $R_{22} = 5k$ (potentiometer), $R_2 = R_{21} + R_{22} = 6K\Omega(\max)$and $1K\Omega(\min)$

$G_1 = \frac{R_3}{R_{21}+R_{22}} + 1 = \frac{100k}{6k} + 1$ to $\frac{100k}{1k} + 1$

= 17.7 to 100

Hence G_{1min} = 17.7 to G_{1max} = 100

(ii) For the second stage of the first order band pass filter:

We have selected R_1 = 1MΩ, R_2 = 6.2K, R_3 = 100KΩ, C_1 = 1 μF, C2= 0.01uF

$f_{c21} = \frac{1}{2\pi \cdot R_1 \cdot C_1} = \frac{1}{2\pi \cdot 1M\Omega \cdot 1\mu F} = 0.159 = 0.2$ Hz, where we have selected f_{c21} as the cut-off frequency of the second stage high pass filter

$f_{c22} = \frac{1}{2\pi \cdot R_3 \cdot C_2} = \frac{1}{2\pi \cdot 100K\Omega \cdot 0.01\mu F} = 159.2$ Hz, where we have selected f_{c22} as the cut-off frequency of the second stage low pass filter

$G_2 = \frac{V_0}{V_i} = \frac{R_3}{R_2} + 1,$

Where we have selected G_2 as the gain of the second stage amplifier

$= \frac{100k}{6.2k} + 1$

$= 17.1$

(iii) 2^{nd} order band pass filter:

Hence the cut-off frequency f_{cH} for the passive 2^{nd} order high pass filter, which is composed of two identical 1^{st} order passive high pass filter with cut-off frequency, $f_{11} = f_{21} = 0.159$ Hz is

$f_{cH} = 1.555 \times f_{11} = 1.555 \times 0.159 = 0.247 \cong 0.2$Hz

Hence the cut-off frequency f_{cL} for the active 2^{nd} order high pass filter with composed of two identical 2^{nd} order passive high pass filter with cut-off frequency $f_{12} = f_{22} = 159.2$ is

$f_c = 0.6434 \times f_{12} = 0.6436 \times 159.2 = 102.461 \cong 102.5$ Hz

Total Gain of the pre-amplifier $G_{TMax} = G_{1max} \times G_2 = 100 \times 17.1290 = 1712.9$

$G_{TMin} = G_{1min} \times G_2 = 17.6667 \times 17.1290 = 302.6$

Thus we have designed a pre-amplifier with a variable gain and band pass filter with the cut-off frequencies range from 0.2-102.5Hz.

2.1.4 Total Gain of the Bio-potential amplifier: To amplify the EEG signal to the level sufficient for processing first the pre-amplifier with the gain of G_{IA} is used and then main amplifier which provides variable gain in the range of G_{TMin} to G_{TMax} is used.

Total amplification range provided to the EEG signal by the bio-potential amplifier is calculated as follows:

$G_{min} = G_{IA} \times G_{TMin} = 3070.9$

$G_{max} = G_1 \times G_{TMax} = 17382.7$

The maximum and minimum amplification provided by the bio-potential amplifier to a 50µV sine wave is shown in the figure 5:

Minimum amplitude: **Maximum amplitude:**

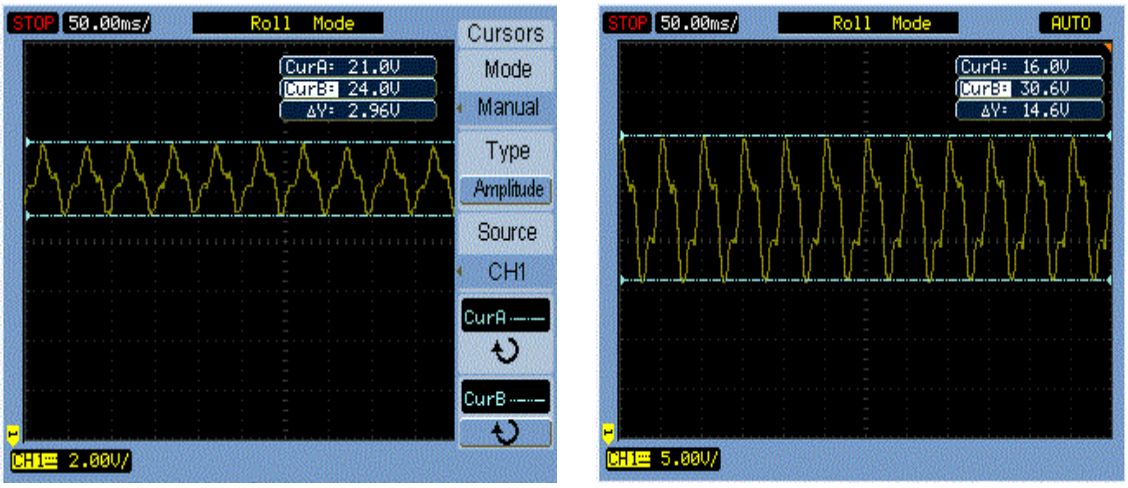


Fig.5. Minimum and the maximum amplification provided by the bio-potential amplifier to a 50µ*V* signal generated from the calibrator and displayed on the oscilloscope.

2.1.5 Driven Right Leg (DRL) circuit: The common mode voltage is further minimized by the DRL circuit by attaching a third electrode to the patient which provides a low-impedance path between the patient and the instrumentation amplifier (IA). The DRL circuit reduces the effective electrode resistance and it only allows a safe amount of current to flow through the third electrode.

The DRL senses the common mode signals found in the instrumentation amplifier are amplifies, inverts and feeds it back to the body via the third electrode. It greatly reduces common mode interference as the signal when feedback to the instrumentation amplifier will cancel common mode noise between the right leg driver (is also called reference electrode) and the electrodes on the brain thus reducing the common-mode noise at the source[45].

The DRL circuit also works as a good amplifier isolator as it introduces large impedance between body and ground achieved by selecting a large register and a small feedback capacitor thus ensuring patient safety.

Thus we have designed a bio-potential amplifier with good amplification ability and high CMRR to eliminate interferences present with the EEG signal, which is suitable for our BCI system.

2.1.6 Electrical Isolation: Optocoupler 6N138/ 6N139is introduced between the Max232 and personal computer (PC) in case of RS232 serial communication. Optocouplers are generally used to transfer data between two devices without making direct electrical connection (electrical isolation). It basically blocks the passage of leakage current from the power line (50-60Hz). If the patient comes in contact with the 120V line, this barrier would prevent dangerous (even fatal) currents from following through the patient by grounding to the ground the microcontroller[46]. It also allows higher transmission rates by reducing the voltage swings.

2.2 Digital Section:

An amplified and conditioned EEG signal enters the digital part which is based on the microcontroller DsPIC30F4013. It consists of analog to digital (A/D) module that converts the EEG signal into a digital signal at the sampling rate of 250Hz. To extract the frequency bands associated with the SSVEP and beta rebound patterns, digital filtering is performed using the DSP engine in the DsPIC30F4013 microcontroller. A prediction algorithm was developed to decode user's intention (amplitude increase in both SSVEP and beta rebound based methods in response to active concentration by the user on an external visual stimulus and performance of a brisk hand movement (or imagination of the movement) respectively) by calculating average amplitude power and comparing it with preset power threshold every predetermined time interval. And if the average calculated power was greater than the power threshold the light switch was turned ON. The algorithm for the digital section is shown in Fig. 6. The digitized and processed EEG signal is relayed to the computer (Dell Vistro desktop), where it is displayed, analyzed and processed continuously using Matlab toolbox called brain computer interface to virtual reality or BCI2VR [47, 48]. To achieve the successful working of the digital part, we need to configure the A/D module, the timer module, the digital filtering module and the (UART) module (to interface the microcontroller with the computer, where the digitized EEG signal is display and analyzed using the BCI2VR software). The following section discusses the in detail configuration of each module and the algorithm to decode and prediction users intention which control an external switch. The schematic diagram of the digital section is shown in Appendix B. All the sections in the digital part are discussed with the aid of dsPIC30F family reference manual, dsPIC30F4013 data sheet and dsPIC Language Tools Libraries[49-51].

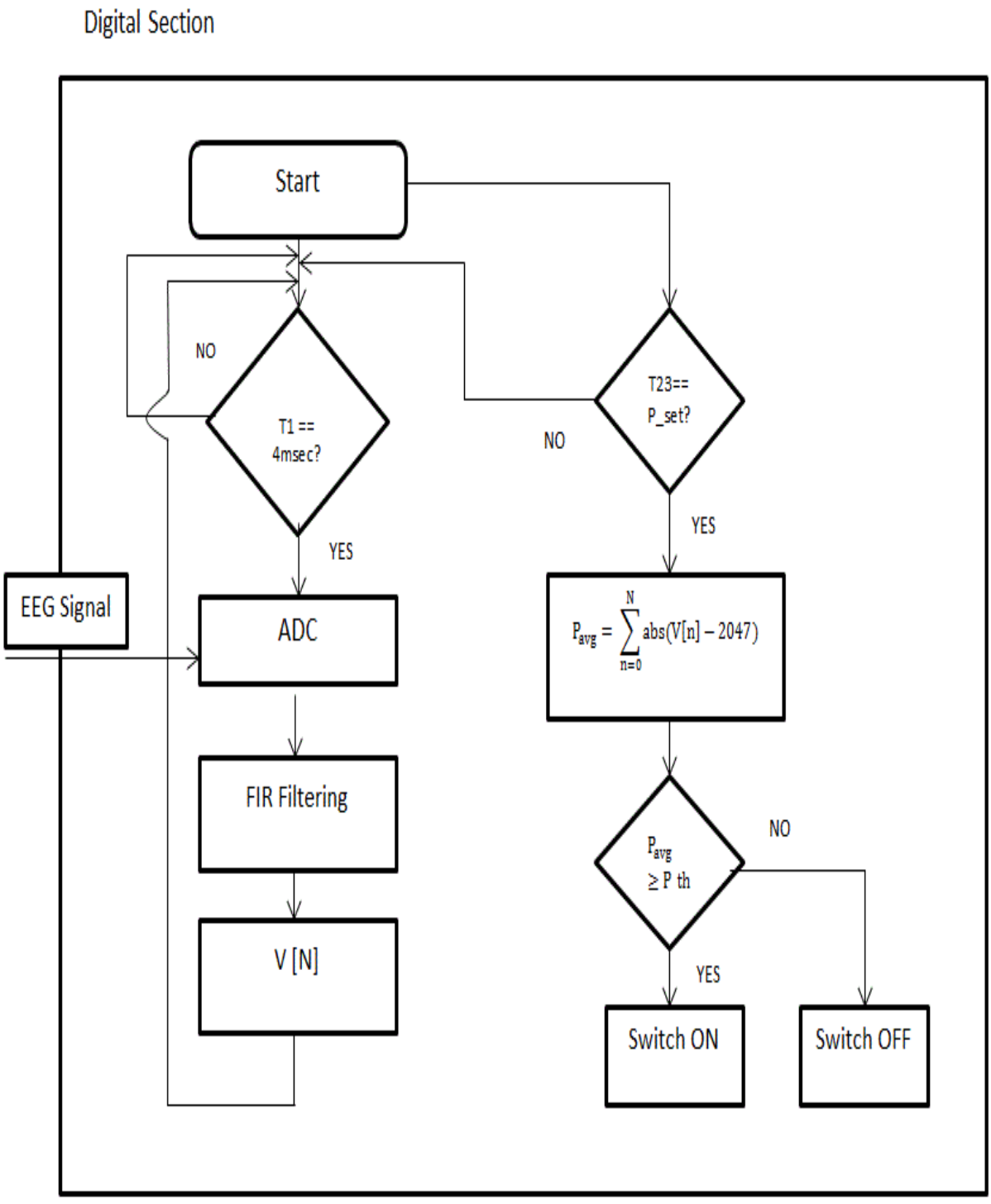


Fig.6. Flow Chart of the Digital Section.

2.2.1 Interrupts and Timers: For our project we are using 16-bit Timer1 to generate an interrupt every 4 ms (which corresponds to the sampling rate of 250Hz) and a 32-bit Timer23 to carry out power calculations every preset interval.

A microcontroller can serve several peripheral devices (such as the UART module, analog to digital converter (ADC) modules, external light emitting diodes(LEDs) etc. not all at the same time) without getting tied down, by using interrupts. An interrupt as the name suggests, is an external or internal event that interrupts the normal execution of central processing unit (CPU) and informs that the microcontroller that its device needs immediate service. Every interrupt has a program associated with it called the interrupt service routine (ISR) or interrupt handler.

(a) When an Interrupt Occurs:

Microcontroller finish's the current instruction it's executing and saves the address of the next instruction on the top of stack. It jumps to the interrupt vector table (group of fixed memory locations set aside to hold the addresses of ISRs). The interrupt vector table directs the microcontroller to the address of ISR. Microcontroller executes all the Instructions of the ISR.

After the last instruction is executed, microcontroller returns to the place where it was interrupted and starts execution from that address.

A DsPIC30F4013 microcontroller has up to 41 interrupt sources and 4 processor exceptions (traps). DsPIC30F microcontrollers have seven priority levels which can be assigned to peripheral interrupt source by writing to the control bits in the appropriate interrupt priority Control register (IPCx, where x denotes the register number, x =1-11). With seven is the highest priority level and one is the lowest level. Assigning a priority level of '0' to an interrupt source is equivalent to disabling that interrupt.

(b) Sources/ Types of interrupts

There are 41 sources of interrupts in the dsPIC30F. Following are some of the most widely used sources of interrupts in the dsPIC30F:

1. There is an interrupt set aside for each of timers (Timer 1, 2, 3, 4 and 5).
2. The ADC interrupts.
3. Peripheral interrupts such as UART transmit and receive, serial peripheral interface (SPI), inter integrated circuit (I2C) etc.
4. Five interrupts INT0-INT4 are used as external hardware interrupts.

(c) To control the interrupt operation we need three bits corresponding to each interrupt they are:

1. Interrupt Flag Status Registers (IFSx (where x denotes the register number, x = 0, 1, and 2)): When an interrupt is activated, the IFSx (interrupt flag) bit is raised. Thus it indicates that an interrupt event occurred.
2. Interrupt Enable Control Registers (IECx (where x denotes the register number, x = 0, 1, and 2)): It allows the programmer to enable (unmask) or disable (mask) corresponding interrupt.
3. Interrupt Priority Control Registers (IPCx (where x denotes the register number, x = 0-11)): Each user interrupt source can be assigned to one of eight priority levels. The IPC registers are used to set the interrupt priority level for each source of interrupt.

   The interrupt priority (IP) bit along with the interrupt flag (IF) and interrupt enable (IE) bits will complete all the flags needed to program the interrupt for the dsPIC30F.

INTCON1, INTCON2 registers control the functions of the global interrupts. INTCON1 register contains control and status flags for the processor trap sources. It also handles the interrupt nesting (for a multi-level priority system, any ISR in progress will be interrupted by another higher user assigned priority level source). Interrupt nesting can be disabled by setting the NSTDIS control bit of the INTCON1 control register. If interrupt nesting is disabled, the user assigned interrupt priority levels are of no use and all the servicing interrupts will be set to the highest priority level(7). User assigned interrupt priority levels will only be used to decide which of the simultaneously pending interrupts will be serviced next. INTCON2 register controls the external interrupts.

Timer in simplified terms is a register, whose value keeps on increasing (or decreasing) on every clock pulse, till it reaches its maximum value and rolls over to generate an interrupt. Thus the timer is used to generate time delay independent of the CPU.

For a timer with internal clock source, 1/ 4^{th} of the frequency of the crystal oscillator on the pins OSC1 and OSC2 of the microcontroller (system clock frequency (F_{osc})/4) is fed into the timer, where system clock period (T_{osc}) = ($4/F_{osc}$). The internal clock of the CPU can be as fast as 120 MHz which is too fast for the timer to operate; therefore a prescaler (P) could be used to decrease the frequency of the Timer.

For example prescale value =1: P, the Timer increment Frequency = ($F_{osc}/4$)× (1/P).

DsPIC30F4013 is equipped with five 16-bit timers (Timer1, Timer2, Timer 3, Timer 4 and Timer 5). Each module consists of the three readable/writeable registers:

1. Timer register (TMRx (16-bit timer count registers, where x =1, 2, 3, 4, and 5)): It is a writable register which holds the start value from which the timer will start counting.

2. Period register (PRx, which is a 16-bit period register associated with the timer): It is a writable register which holds the final value of the count. TMRx increments on every instruction cycle until it reaches the value preloaded in the PRx register, and then it resets (starts counting from zero again). This register increases the execution speed of operation since timer value is checked in hardware rather than in software.
3. Timer x control register (TxCON where x =1-5): It is a 16-bit control register associated with the timer.

Timer overflow interrupt: An overflow occurs when a Timer has counted till its maximum value (or final value loaded in the PRx register). For example, when a 16-bit Timer 1 reaches its maximum count of 65535, the next clock cycle cause the timer to rolls over to 0 (or value loaded in the TMRx register) and start counting again, thus generating an overflow. An overflow can trigger an interrupt (set the Interrupt Flag Status bit (TxIF)) if the associating bits such as Interrupt Enable Control bit (TxIE) and Interrupt Priority Control bits (TxIP<2:0>) of the timer module are set.

Steps to design Timer1 to interrupt every 4msec (or at a sampling rate of 250 Hz) are as follows:

1. Configuration of Timer1: The Timer 1 control register (T1CON) is shown in Fig. 7.and the description of T1CON registers bits in discussed in Table 1.

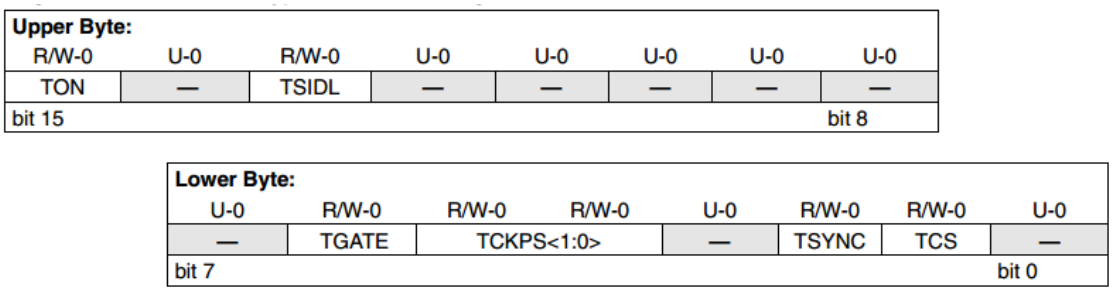


Fig.7. Timer 1 control register.

Table 1. Timer 1 control (T1CON) register bits description

T1CON bit	Description
bit 15	TON: Timer On Control bit 1 = Starts the timer 0 = Stops the timer
bit 14	Unimplemented: Read as '0'
bit 13	TSIDL: Stop in Idle Mode bit (When the processor enters Idle mode, its system clock is functional but the CPU stops code execution) 1 = Discontinue timer operation when device enters Idle mode 0 = Continue timer operation in Idle mode
bit 12-7	Unimplemented: Read as '0'
bit 6	TGATE: Timer Gated Time Accumulation Enable bit, in this mode the internal timer register increments till the state of the T1CK (external clock) pin is high. T1IF interrupt flag will be set whenever the timer counts up to value preloaded in the PR1 register or the T1CK pin state is changed to low. 1 = Gated time accumulation enabled 0 = Gated time accumulation disabled

bit 5-4 TCKPS<1:0>: Timer Input Clock Prescale Select bits, Timer1 has the following prescale option

11 = 1:256 prescale value

10 = 1:64 prescale value

01 = 1:8 prescale value

00 = 1:1 prescale value

bit 3 Unimplemented: Read as ‘0’

bit 2 TSYNC: Timer External Clock Input Synchronization Select bit, used specifically for external clock operation as a counter

When TCS = 1:

1 = Synchronize external clock input

0 = Do not synchronize external clock input

When TCS = 0:

This bit is ignored. Read as ‘0’. Timer1 uses the internal clock when TCS = 0.

bit 1 TCS: Timer Clock Source Select bit, selects whether the clock for the instruction cycle will be from internal or external clock

1 = External clock from pin TxCK

0 = Internal clock (FOSC/4)

bit 0 Unimplemented: Read as ‘0’

To Configure Timer1 Module for our application, we load the T1CON register with the following value:

T1CON = 0xA030 or 0b1010000000110000; Start timer, use the internal clock ($F_{osc}/4$) and prescale the clock value to 256

Hence the timer increment frequency in our case is:

Crystal oscillator = 10MHz, PLL =8, F_{osc} = 10MHz x 8=80MHz

Instruction cycle frequency ($F_{CY)}$ = ($F_{osc}/4$) = 20MHz,

million instructions per second (MIPS) = 20;

Timer increment frequency = ($F_{osc}/4$) × (1/Prescalar) = (20MHz × 1/256) = 78125Hz

Or equivalently we can use the dsPIC peripheral library timer functions (with the inclusion of timer.h header) to configure the above register in the following way:

```
OpenTimer1 (T1_ON & T1_GATE_OFF & T1_IDLE_STOP &
  T1_PS_1_64 & T1_SYNC_EXT_OFF &
      T1_SOURCE_INT, match_value);
```

2. Load the value in the Timer1 period register PR1, if we know the amount of delay required we can calculate the values and load it into the PR1 register as follows:

   PR1 = Timer increment frequency x Amount of delay

   For example to generate a delay of 4 ms= 78125 x 4 ms= 312.5 = 0x0139

   Hence PR1 = 0x0139 or match_value = 0x0139.

3. Load the start value in the TMR1 register. TMR1=0x0000 or WriteTimer1 (0x0000).
4. Enable the Timer1 overflow interrupt, set the Interrupt priority and clear Timer1 interrupt flagIEC0bits.T1IE = 1, IFS0bits.T1IF=0, IPC0bits.T1IP<2:0> =110 (6) or we can use timer function ConfigIntTimer1 (T1_INT_PRIOR_6 & T1_INT_ON).

5. As soon as the timer counts up to the preloaded value of PR1, Timer1 overflows and the Timer1 interrupt flag bit is set (IFS0bits.T1IF=1). A microcontroller looks up the address in the interrupt vector table and executes the ISR associated with the Timer. Before exiting the ISR the Timer overflow flag bit needs to be cleared (IFS0bits.T1IF=0). Also before exiting the ISR we need to reload the initial count values in the Timer registers (in our case WriteTimer1(0x0000)).

Steps to design Timer23 to calculate the power value every preset time interval is as follows:

We are using 32-bit timer to provide4 sec synchronization signal (go-signal) via a LED (which remains ON for 3 sec and OFF for 1 sec) and to calculate the average amplitude power every 4 second for beta rebound method. For SSVEP based method we use the 32-bit timer to calculate the average amplitude power used to regulate the switch every 0.5 second. The steps to design the timer to work in 32-bit mode are as follows:

1. Configuration of a 32-bit timer module is formed by combining two 16-bit Timer2 and Timer3modules. When configured for 32-bit operation, control bits of Timer 2 (T2CON control register, PR2 period register and TMR2 register) control the operation of 32-bit timer (where T3CON of timer 3 has no effect). And for interrupt control, control bits of Timer3 register (IFS0bits.T3IF, IEC0bits.T3IE and IPC1bits.T3IP<2:0>) are used. The Timer 2 control register (T2CON) is shown in Fig.8. and the description of T2CON registers bits in discussed in Table 2.

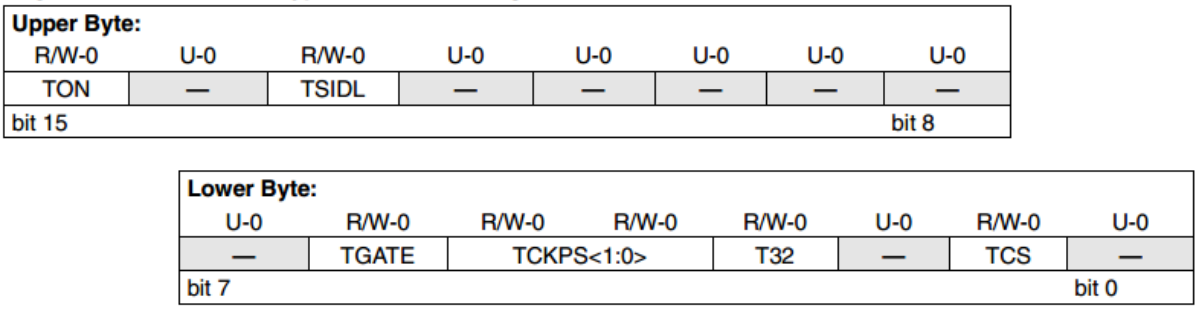


Fig.8. Timer 2 control register.

Table 2: Timer 2 control (T2CON) register bits description

T2CON bits	Description
bit 15	TON: Timer On bit When T32 = 1 (in 32-bit Timer mode): 1 = Starts 32-bit TMRx:TMRy timer pair 0 = Stops 32-bit TMRx:TMRy timer pair When T32 = 0 (in 16-bit Timer mode): 1 = Starts 16-bit timer 0 = Stops 16-bit timer
bit 14	Unimplemented: Read as '0'
bit 13	TSIDL: Stop in Idle Mode bit (same as Timer1) 1 = Discontinue timer operation when device enters Idle mode 0 = Continue timer operation in Idle mode
bit 12-7	Unimplemented: Read as '0'
bit 6	TGATE: Timer Gated Time Accumulation Enable bit (same as Timer1)

1 = Timer gated time accumulation enabled

0 = Timer gated time accumulation disabled

(TCS must be set to logic ‘0’ when TGATE = 1)

bit 5-4 TCKPS<1:0>: Timer Input Clock Prescale Select bits

11 = 1:256 prescale value

10 = 1:64 prescale value

01 = 1:8 prescale value

00 = 1:1 prescale value

bit 3 T32: 32-bit Timer Mode Select bits, determines whether the timer is operating in 32-bit mode or 16-bit mode.

1 = TMRx and TMRy form a 32-bit timer

0 = TMRx and TMRy form separate 16-bit timer

bit 2 Unimplemented: Read as ‘0’

bit 1 TCS: Timer Clock Source Select bit

1 = External clock from pin TxCK

0 = Internal clock (FOSC/4)

bit 0 Unimplemented: Read as ‘0’

To Configure Timer2 Module for our application we have loaded the T2CON register with following value:

T2CON = 0xA02A or 0b1010000000101010; Start timer, use the internal clock (F_{osc}/4) and prescale the clock value to 64

Hence the timer increment frequency in our case is:

Crystal oscillator = 10MHz, PLL =8, F_{osc} = 10MHz x 8=80MHz

F_{cy} = (F_{osc}/4) = 20MHz, MIPS = 20;

Timer increment frequency = (F_{osc}/4) × (1/Prescalar) = (20MHz × 1/64) = 312500Hz

Or equivalently we can use the dsPIC peripheral library Timer functions (with the inclusion of timer.h header) to configure the above register in the following way:

```
OpenTimer23(T2_ON & T2_GATE_OFF & T2_IDLE_STOP &
            T2_PS_1_64 & T2_32BIT_MODE_ON &
            T2_SYNC_EXT_OFF &
            T2_SOURCE_INT, match_value);
```

2. Load the value in the period register of 32-bit timers, PR3 register holds the most significant word (MSWord) and PR2 register holds the least significant word (LSWord). The maximum value that can be loaded into the PR3:PR2 pair is 0xFFFFFFFF or 4294967295. In this case we load in the max value (PR2 = 0xFFFF and PR3 =0xFFFF or match_value =0xFFFFFFFF)
3. Load the start value in the TMR3:TMR2 register (similarly here TMR3 holds the MSWord and TMR2 holds the LSWord). To write, user must first write the MSWord data to the TMR3HLD register (used in the 32-bit operation, which will automatically transfer TMR3HLD contents into the TMR3 register) and then write the LSWord contents into TMR2 register.

   If we know the amount of delay required we can calculate the values and load it into the TMR register as follows:

   For example to generate a delay = Timer increment frequency x (Amount of delay)

3.1. For beta rebound based method:

i. For the LED to be ON for3 second and to calculate power after every 4 sec, timer increment frequency is 312500 × 3 = 937500

Timer32-bit start value = 4294967296 - 937500 = 4294029796 = 0xFFF1B1E4, which is loaded in the TMR3:TMR2 register as follows:

TMR3HLD = 0xFFF1;

TMR2 = 0xB1E4;

Or WriteTimer23 (0xFFF8D8F2); using the dsPIC30F libraries

ii. For the LED to be OFF for 1 second, timer increment frequency is 312500 × 1 = 312500

Timer 32-bit start value = 4294967296- 312500 = 4294654796 = 0xFFFB3B4C

Similarly we load timer23 with this value as WriteTimer23 (0xFFFB3B4C)

3.2. For the SSVEP based method:

To calculation power after every 0.5 second we program an interrupt to generate every 0.5 second, timer increment frequency is 312500 × 0.5 = 156250

Timer 32-bit start value = 4294967296-156250 = 4294811046 = 0xFFFD9DA6

Similarly we load timer23 with this value as WriteTimer23 (0xFFFD9DA6);

4. Enable the Timer3 overflow interrupt, set the Interrupt priority and clear Timer3 interrupt flagIEC0bits.T3IE = 1, IFS0bits.T3IF =0, IPC1bits.T3IP<2:0> = 111 (priority 7) or we can use Timer function ConfigIntTimer23 (T3_INT_PRIOR_7 & T3_INT_ON);
5. As soon as the timer counts up to the preloaded value of PR3:PR2 = 0xFFFFFFFF, 32-bit Timer overflows and interrupt flag bit of TMR3 is set (IFS0bits.T3IF=1). A microcontroller looks up the address in the interrupt vector table and executes the ISR associated with the

Timer3. Before exiting the ISR the Timer overflow flag bit needs to be cleared (IFS0bits.T3IF=0). Also before exiting the ISR we need to alternately reload the initial count values WriteTimer23 (0xFFF8D8F2) to get the delay of 1.5 second and WriteTimer23 (0xFFF4143F) to get the delay of 2.5second.

2.2.2 Universal Asynchronous Receiver Transmitter (UART): We are using the UART module to interface our BCS device to the PC for offline processing, steps to interface the UART module with PC are discussed in Appendix F. UART is a full-duplex asynchronous system used for serial communication (data is sent one bit at a time) with peripheral devices (devices with a serial port such as personal computers, RS232 and RS-485 interfaces). DsPIC30F4013 has two UART modules (UxMODE) built-in making data transfers faster and cheaper. The rate of data transfer in serial data communication is stated in bits per second (bps). Another widely used terminology for bps is baud rate. The important features of UART module in dsPIC30F family is it uses standard non-return-to-zero(NZR) format (one start bit, eight or nine data bits, and one or two stop bits), parity options (for 8-bit data), built-in baud rate generator, error detection capabilities, separate transmit and receive interrupts and loopback mode for diagnostic support.

RS232: RS232 is most widely used asynchronous serial communication protocol used in computers and digital systems. The DsPIC30F4013 has four pins that are used specifically for transferring and receiving data serially. These four pins are called U1ATX, U1ARX, U2TX and U2RX. These pins use TTL compatible logic levels (High (1) =+5 V and low (0)) = 0V), however in RS232 High (1) = 12V and LOW=+12V (this is done to increase the range and reliability of the data transfer). To facilitate the interface a line drive (voltage converter) needs to be used. MAX232 is a line drive used for this project.

MAX232: The MAX232 converts RS232 voltage levels to TTL voltage levels, and vice versa. Four capacitors are used for voltage conversions. MAX232 has two sets of line drivers for transferring and receiving data. The line drivers used for transmission (TX) pins are called T1 and T2, while the line drivers for reception (RX) pins are designated as R1 and R2. In many applications only one of each is used. For example, T1 and R1 are used together for TX and RX pins of dsPIC30F4013 and second one is left unused. The circuit connections of the UART module in the dsPIC30F4013 microcontroller to the MAX232 and PC are shown in Appendix B.

In dsPIC30F4013, Each UART module is equipped with a 9-bit wide FIFO transmit data buffer called UARTX transmit register (UxTXREG) which can hold up to 4 words. A UART module (UART1 or UART2) can only be enabled by setting the UART enable bit (UARTEN) and UTXEN bits of the UARTx Mode Register (UxMODE) and UARTX status and control Register (UxSTA, where x = 1 or 2) registers (where x corresponds to UART1 or UART2), along with configuring its TX and RX pins as output and input respectively). However the actual transmission won't occur until the UxTXREG register is loaded with data and baud rate generator produces the baud rate. We are using UART2's transmit mode to display our digitized results on the PC.

Steps to design UART2 module as a transmitter are as follows:

1. Configure the UxTX and UxRX pins of the UxMODE (where x is 1 or 2) as inputs and outputs

   -UxRX is configured as input (For UART2, its TX and RX pins are multiplexed with port F hence we configure TRISFbits.TRISRF4=1).

   -TX is configured as output (For UART2, TRISFbits.TRISRF3 =0).

2. Configuring the Control Registers or setting up the UART

2.1 The configuration of the UART mode control register is shown in figure 9 and the description of the bits are shown in table 3.

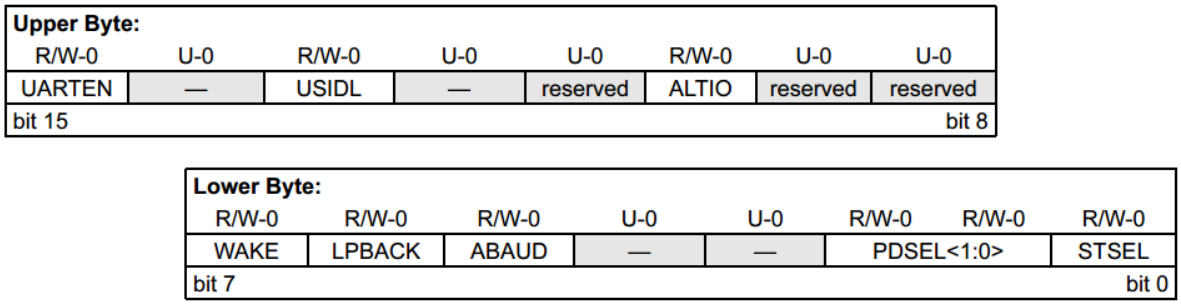


Fig.9. UARTx Mode control (UxMode) register.

Table 3: UARTx Mode control register bits description

UxMODE bits	Description
bit 15	UARTEN: UART Enable bit 1 = UART is enabled. 0 = UART is disabled.
bit 14	Unimplemented: Read as '0'
bit 13	USIDL: bit determines whether the module will stop or continue normal operation in Idle mode 1 = Discontinue operation when device enters Idle mode 0 = Continue operation in Idle mode
bit 12	Unimplemented: Read as '0'
bit 11	Reserved: Write '0' to this location

bit 10 ALTIO: UART Alternate I/O Selection bit, the alternate UART pins are beneficial when the primary UART pins are shared by other peripherals.

1 = UART communicates using UxATX and UxARX I/O pins

0 = UART communicates using UxTX and UxRX I/O pin

bit 9-8 Reserved: Write ‘0’ to these locations

bit 7 WAKE: Enable Wake-up on Start is used to wake the dsPIC from the sleep mode is only beneficial during the reception mode.

1 = Wake-up enabled

0 = Wake-up disabled

bit 6 LPBACK: UART Loopback Mode Select bit

1 = Enable Loopback mode, In this mode the UxRX pin is disconnected from the UART receive logic and connected internally to the UxTX pin.

0 = Loopback mode is disabled

bit 5 ABAUD: Auto Baud Enable bit is useful in the reception mode to determine the baud rates of the received characters

1 = Input to Capture module from UxRX pin

0 = Input to Capture module from ICx pin

bit 4-3 Unimplemented: Read as ‘0’

bit 2-1 PDSEL<1:0>: Parity and Data Selection bits are used to select the data length and parity used in the transmission

11 = 9-bit data, no parity

10 = 8-bit data, odd parity

01 = 8-bit data, even parity

	00 = 8-bit data, no parity
bit 0	STSEL: Stop Selection bit determines the number of stop bits used during the data transmission 1 = 2 Stop bits, 0 = 1 Stop bit

2.2 Configuration of the UARTx status and control register (UxSTA) is shown in Fig. 10.and the description of T1CON registers bits in discussed in Table 4.

Upper Byte:

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Fig.10. UARTx Status and control register.

Table 4. UARTx Status and control register (UxSTA, where x=1 or 2) bits description

UxSTA bits	Description
bit 15	UTXISEL: Transmission Interrupt Mode Selection bit 1 = Interrupt is generated when a word is transferred from Transmit buffer to the Transmit Shift register (UxTSR) and as result, the transmit buffer becomes empty. 0 = Interrupt is generated when all four words are transferred from the Transmit buffer to the Transmit Shift register
bit 14-12	Unimplemented: Read as ‘0’

bit 11 UTXBRK: Transmit Break bit drives the UxTX line to ‘0’. This bit is set in software and runs for 13 baud clocks. This needs to be cleared in software where it generates a stop bit after which the transmitter can renew its activity

1 = UxTX pin is driven low, regardless of transmitter state

0 = UxTX pin operates normally

bit 10 UTXEN: Transmit Enable bit, This bit finally enables the UART transmission and also sets the UxTXIF flag bit if UTXISEL = 0

1 = UART transmitter enabled.

0 = UART transmitter disabled, any pending transmission is aborted and buffer is reset.

Note: The UTXEN bit should not be set until the UARTEN bit has been set. Otherwise, UART transmissions will not be enabled.

bit 9 UTXBF: Transmit Buffer Full Status bit (Read Only), Each UART module is equipped with a 9-bit wide FIFO transmit data buffer which can hold up to 4 words. UTXBF status bit is set if the buffer is full

1 = Transmit buffer is full, no new data will be accepted in the FIFO buffer

0 = Transmit buffer is not full, at least one more data word can be written

bit 8 TRMT: Transmit Shift Register is Empty bit (Read Only) is used to check whether the last transmission has completed

1 = Transmit shift register is empty and transmit buffer is empty

0 = Transmit shift register is not empty, a transmission is in progress or queued in the transmit buffer

bit 7-6 URXISEL<1:0>: Receive Interrupt Mode Selection bit used for UART

Reception hence Don’t Care(‘00’)

bit 5 ADDEN: Address Character Detect (bit 8 of received data = 1) used for UART Reception hence Don’t Care(‘0’)

bit 4 RIDLE: Receiver Idle bit (Read Only) used for UART Reception hence Don’t Care(‘0’)

bit 3 PERR: Parity Error Status bit (Read Only) used for UART Reception hence Don’t Care(‘0’)

bit 2 FERR: Framing Error Status bit (Read Only) used for UART Reception hence Don’t Care(‘0’)

bit 1 OERR: Receive Buffer Overrun Error Status bit (Read/Clear Only) used for UART Reception hence Don’t Care(‘0’)

bit 0 URXDA: Receive Buffer Data Available bit (Read Only) used for UART Reception hence Don’t Care(‘0’)

2.3. Baud Rate Generator (BRG) is a 16-bit readable and writable register used to allow maximum flexibility in baud rate generation.

Formula to Calculate Baud Rate is as follows:

Baud Rate = $\frac{F_{CY}}{(16*(BRG+1))}$

Where, BRG is a 16-bit value held in the BRG register (0-65535, Hence the maximum baud rate possible is $\frac{F_{cy}}{16}$ and the minimum rate is $\frac{F_{cy}}{16\times65536}$), F_{cy} is the instruction clock rate

Calculation of UART2 Baud Rate with BRGH = 1 for F_{cy} of 20MHz (20 MIPS) and desired baud rate 115200

$$\text{UxBRG} = \frac{F_{CY}}{(16 \times \text{Baud Rate}))} - 1$$

$$= \frac{20Mhz}{(16 \times 15200))} - 1$$

$$= 9.85 \cong 10$$

Hence U2BRG =10;

Calculated Baud Rate =113636.36

$$\text{Error} = \frac{((\text{Calculated Baud Rate} - \text{Desired Baud Rate}))}{(\text{Desired Baud Rate})}$$

$$= \frac{((113636.36 - 115200))}{(115200)}$$

= -1.2%

For F_{cy} = 20MHz, (F_{osc} = (10MHz x 8 (PLL))) other working baud rates are shown in Table 5.

Table 5 BRG values at different Baud Rates for a 20MHz crystal oscillator

Baud Rate (bps)	BRG value	Error (%)
9600	129	0.2
19200	64	0.2
57600	21	1.5
115200	10	-1.2

Configuration of the UART2 Module for our application:

UART2 Mode Register, U2MODE = 0xC000 or 0b1010000000000000; to select the data format of 8-bits, no parity and one stop bit.

U2STA = 0x8500 or 0b1000010100000000; to finally enable the UART2 transmission, generate an interrupt when all four words are transferred from the Transmit buffer to the Transmit Shift register,

U2BRG = 10

Or equivalently we can use the dsPIC peripheral library to configure the above register in the following way:

unsigned int U2MODEvalue, U2STA, Baud;

U2MODEvalue = UART_EN & UART_IDLE_STOP &UART_DIS_WAKE &UART_DIS_LOOPBACK &UART_DIS_ABAUD & UART_NO_PAR_8BIT & UART_1STOPBIT;

U2STAvalue = UART_INT_TX_BUF_EMPTY & UART_TX_PIN_NORMAL & UART_TX_ENABLE & UART_INT_RX_3_4_FUL & UART_ADR_DETECT_DIS & UART_RX_OVERRUN_CLEAR;

Baud =10

OpenUART2 (U2MODEvalue, U2STAvalue, Baud);

3. Monitor the TRMT bit of the U2STA (or UxSTA) register to make sure last transmission has completed (or the UART is not transmitting) and is ready for next byte. We use while(BusyUART2()) function of the dsPIC peripheral library to achieve this, which is similar to checking if (U2STAbits.TMRT ==1).

4. The character byte (255) to be transmitted serially is written into the U2TXREG register. Configuration of UARTx transmit register (UARTx) is shown in Fig.11. and the description of T1CON registers bits in discussed in Table 6.

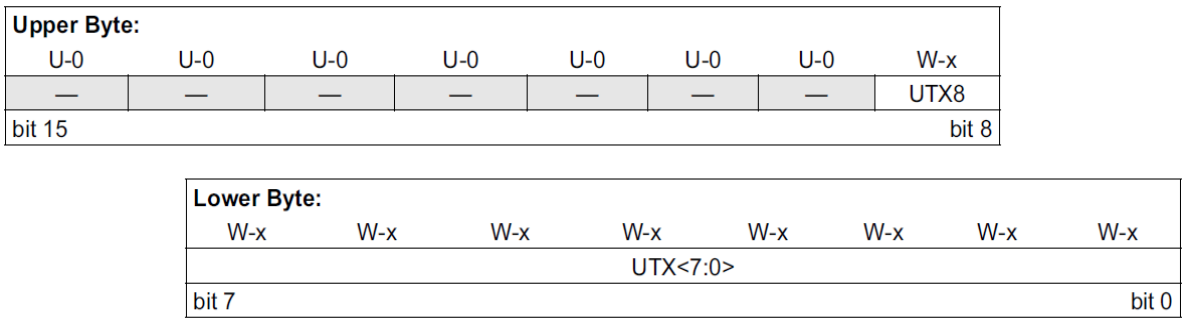


Fig.11. UARTx transmit register (Write Only), where x =1 or 2.

Table 6. UARTx transmit register bits description, where x = 1 or 2

UARTx bits	Description
bit 15-9	Unimplemented: Read as '0'
bit 8	UTX8: Data bit 8 of the Character to be Transmitted (in 9-bit mode)
bit 7-0	UTX<7:0>: Data bits 7-0 of the Character to be Transmitted

As state above UxTXREG is 9-bit FIF0 buffer which can store four words. The UxTXREG register is loaded with data in software (For example U2TXREG = 0x5A). The heart of the transmitter is the Transmit Shift register (UxTSR) its data from UxTXREG. This UxTSR at every shift clock (Baud rate bps) generated from the baud rate generator transmits each bit serially. Normally when transmission is first started, the UxTSR register is empty, so a transfer to the UxTXREG register will result in an immediate transfer to UxTSR. We have used WriteUART2 (0x5A); to achieve this.

5. To transfer the next character, go to step 3.

2.2.3. Analog to Digital Converter (ADC): We are using the ADC module in the dsPIC30F4013 to convert analog EEG signals to digital signals for processing. Real world signals are analog (continuous) in nature. For example to measure every day physical quantity such as temperature, a temperature sensor is used to convert it into an electrical signal (voltage). But as our microcontroller is digital (discrete) in nature, we need ADC to translate the analog signals as digital numbers so that the microcontroller could read and process them.

DsPIC30F4013 contains 12-bit Analog to digital converter (A/D) with 16 analog input channels (AN0-AN15, plus two analog inputs for external voltage reference connections (to set up voltage range other than 0-5V (that isV_{REF+} and V_{REF-} we can set up voltage range of 3.3 to 0.5V))) that can accurately measure voltages and convert it corresponding digital format. The whole system is based around a dsPIC30F microcontroller which is clocked at 80 MHz crystal (20MIPS (million instructions per sec) execution speed). The ADC module is based on Successive Approximation Register (SAR) architecture. A/D module has six 16-bit registers for its configuration and operation they are the ADCON1, ADCON2 and ADCON3 control registers, the ADCHS register to select the input pins to be connected to the S/H amplifiers, ADPCFG register to configure the analog input pins. The ADCSSL register to select inputs to be sequentially scanned.

The 16 available analog inputs are connected to the Sample/Hold amplifier (or S/H channel) designated CH0 which samples the analog input pin voltage, via multiplexer, which is in turn connected to the conversion logic generating results. The S/H amplifier can scan all the 16

analog channels in one A/D operation if the Analog Input Scan Mode is enabled. Results of the ADC conversion is stored in a 16-word result buffer (RAM) designated ADCBUF0-ADCBUF15. We are using an analog channel for the digitization of EEG.

The most important factor in judging the performance of an ADC is its resolution; it specifies how accurately the ADC measures the analog input signals. Higher the resolution smaller is the step size. Therefore for an ADC with 12 bit resolution and the analog input range from 0-5 volts, the Step size can be calculated as follows:

Step Size $=\frac{V_{Ref}}{2^n}$ where n is a ADC with n-bit resolution

$=\frac{5}{4096}$ = 1.22 mV approximately, thus the ADC can be measured accurately up to 1.22mV.

The output voltage from the A/D conversion is calculated as follows:

$$D_{out} = \frac{V_{in}}{\text{Step Size}}$$

Where D_{out} =Digital output voltage (Decimal), V_{in} = Analog input voltage.

Working of the A/D Conversion module: Analog to digital conversion is performed by the A/D conversion module by carrying out the following operations. The A/D module's S/H amplifier is connected to the analog input pin to take a sample of the analog input and hold that sample for analysis. The total sampling time for the A/D is a function of the internal amplifier settling time and the holding capacitor charging time. For an accurate A/D converter the charge holding capacitor should be able to fully charge to the voltage level on the analog input pin which is also defined as A/D acquisition time equivalent to $1T_{AD}$ (defined as the A/D conversion time per bit). Microchip specifies for correct A/D conversions of dsPIC30F, minimumT_{AD} time should be

around 667 nanoseconds. The sampling time is ended manually by clearing the SAMP control bit in the user software or automatically by triggering conversion source (which will terminate acquisition and start the analog to digital conversions). The A/D converter module is disconnected from the end of the analog input pin at the end of sample time. Conversion time is the time required for the A/D converter to convert the voltage held by the S/H amplifier. The A/D converter requires one A/D clock cycle (TAD) to convert each bit of the result. A total of 14 TAD cycles are required to perform the complete conversion.

A/D Conversion Clock $T_{AD} = T_{CY} \times (0.5 \times (ADCS < 5:0 > +1))$

where ADCS <5:0> are bits of the ADCON3 register.

When the conversion is complete, the result is loaded into ADCBUF buffer register and the S/H can be reconnected to the analog input pin. Also, the total conversion time is the sum of acquisition time and the A/D Conversion time which is equal to 15 T_{AD} for 12-bit A/D conversion. We use one channel (AN12) of 12-bit A/D converter, to digitize the analog EEG signal.

Steps to design single Channel A/D module are as follows:

1. The ADCON1, ADCON2 and ADCON3 registers control the operation of the A/D module information and the working of the registers is discussed below.

   1.1 Configuration of A/D control register 1 (ADCON1) is shown in Fig.11. and the description of T1CON registers bits in discussed in Table 7.

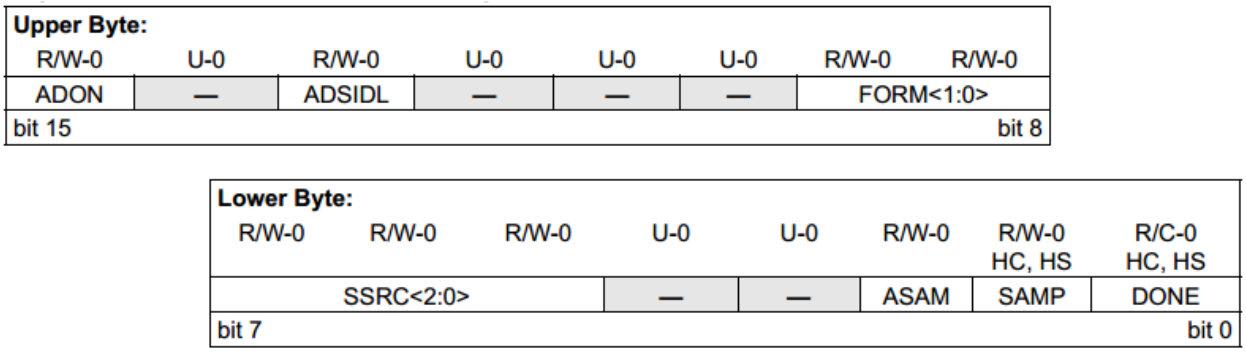


Fig.11. ADCON1: A/D Control Register 1.

Table 7 A/D Control register 1 bits description

ADCON1 bits	Description
bit 15	ADON: A/D Operating Mode bit will turn ON the A/D Conversion module. 1 = A/D converter module is operating 0 = A/D converter is off
bit 14	Unimplemented: Read as '0'
bit 13	ADSIDL: Stop in Idle Mode bit 1 = A/D Module will stop in IDLE mode and abort all current conversions. When the device resumes normal operation partially completed conversions will be ignored 0 = A/D module will continue normal operation in Idle mode. If an A/D interrupt occurs the devices will wake-up from the Idle mode and will resume its execution in the A/D ISR or from next instruction (after the instruction which placed the device in the Idle mode).

bit 12-10 Unimplemented: Read as ‘0’

bit 9-8 FORM<1:0>: Data Output Format bits the output should be programmed in one of the following formats where s is the sign bit

11 = Signed fractional (DOUT = sddd dddd dddd 0000)

10 = Fractional (DOUT = dddd dddd dddd 0000)

01 = Signed integer (DOUT = ssss sddd dddd dddd),

00 = Integer (DOUT = 0000 dddd dddd dddd)

For example: 4096 in following formats

Integer = 0000 1111 1111 1111 = 4095

Signed Integer = 0000 0111 1111 1111 = 2047

Fractional = 1111 1111 1111 0000 = 0.99975 (converter it to decimal and divided by 65536, 65520/65536 = 0.99975)

Signed Fractional format = 0111 1111 1111 0000 =0.9995 (same process as above the result will be 0.499755859375, multiply it by 2 as signed divides the result into half we get 0.99951171875)

In dsPIC30F microcontrollers, digital filtering only accepts input in the signed format in order to convert it into unsigned integer format

UnsignedintResult = (SignedFractionalResult>> 4) + 2048 ; shift the bits four times to match the integer format and add 2048 (since unsigned signed results are from 0-4096 and signed results are from -2047-2048)

bit 7-5 SSRC<2:0>: Conversion Trigger Source Select bits

111 = The conversion trigger source depends on the SAMC bits of

the ADCON3 control register which selects the number of A/D clocks between the start of acquisition and start of conversion.

110 = Reserved

101 = Reserved

100 = Reserved

011 = Motor Control PWM interval ends sampling and starts conversion

010 = General purpose Timer3 compare ends sampling and starts conversion

001 = Active transition on INT0 pin ends sampling and starts conversion

000 = Clearing SAMP bit ends sampling and starts conversion

bit 4-3 Unimplemented: Read as '0'

bit 2 ASAM: A/D Sample Auto-Start bit, Starting of sampling time of ADC input can be controlled automatically by hardware by setting the ASAM bit.

1 = Sampling begins immediately after last conversion completes. SAMP bit is auto set

0 = Sampling begins when SAMP bit set

bit 1 SAMP: A/D Sample Enable bit starts the time for sampling and is controlled by software

1 = At least one A/D sample/hold amplifier is sampling

0 = A/D sample/hold amplifiers are holding

	When ASAM = 0, writing '1' to this bit will start sampling. When SSRC = 000, writing '0' to this bit will end sampling and start conversion.
bit 0	DONE: A/D Conversion Status bit is used to determine whether the A/D conversion is completed. 1 = A/D conversion is done 0 = A/D conversion is not done Clearing this bit will not affect any operation in progress. Cleared by software or start of a new conversion.

For single channel A/D conversion we load ADCON1 with the following value,

ADCON1 = 0x03E0 or 0b0000001111100000; SSRC bit =111 implies internal counter ends sampling and starts converting (T_{AD} based conversion where conversion trigger source depends on the SAMC bits of the ADCON3) and result format as signed fraction since digital filtering in the dsPIC30F microcontrollers accepts filter input in the signed fraction format.

1.2 Configuration of A/D Control Register 2 (ADCON2) shown in Fig.12.and the description of T1CON registers bits in discussed in Table 8.

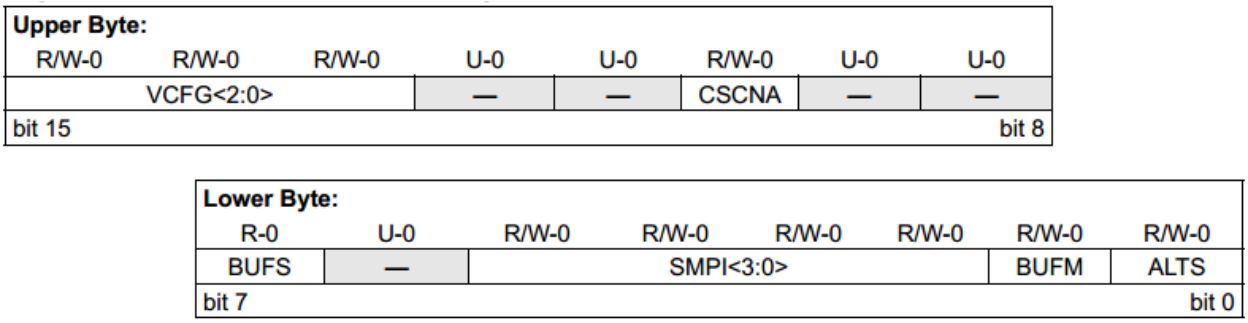


Fig.12. ADCON2: A/D Control Register 2.

Table 8: A/D Control register 2 bits description

ADCON2 bits	Description
bit 15-13	VCFG<2:0>: Voltage Reference Configuration bits are used to select the voltage reference for the A/D module. The upper A/D V_{REF} H and the lower voltages A/D V_{REF} L can be the internal reference voltages or Voltages at pins AN0 and AN1 respectively.
bit 12	Reserved: User should write '0' to this location
bit 11	Unimplemented: Read as '0'
bit 10	CSCNA: Channel 0 S/H has a capability to scan through the selected vector inputs specified by the ADCSSL register by setting the CSCNA bit. When

Bits 15-13	**A/D V_{REF}H**	**A/D V_{REF}L**
000	AV_{DD}	AV_{ss}
001	External V_{REF+} pin	AV_{SS}
010	AV_{DD}	External V_{REF-} pin
011	External V_{REF+} pin	External V_{REF-} pin
1xx	AV_{DD}	AV_{ss}

CSCNA is set, the CH0SA<3:0> bits are ignored.

1 = Scan inputs

0 = Do not scan inputs

bit 9-8 Unimplemented: Read as ‘0’

bit 7 BUFS: Buffer Fill Status bit is used in conjunction with BUFM (if BUFM is set)

1 = A/D converter is currently filling buffer 0x8-0xF, user should read data from 0x0-0x7

0 = Situation is revered and the user should access data in 0x8-0xF

bit 6 Unimplemented: Read as ‘0’

bit 5-2 SMPI<3:0>: Sample/Convert Sequences Per Interrupt Selection bits

1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence

1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence

.....

0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence

0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 BUFM: Buffer Mode Select bit, In the Buffer fill mode 16-word result buffer ADCBUF is split into two8-word buffers ADCBUF (7-0) and ADCBUF (15-8).

1 = Buffer configured as two 8-word buffers ADCBUF (15...8), ADCBUF (7...0), here the two buffers will alternately receive the conversion results after each interrupt event. It has to be used when the processor cannot unload the buffer within the sample and conversion time

0 = Buffer configured as one 16-word buffer ADCBUF(15...0), if the processor can quickly unload a full buffer within the time it takes to sample and convert one channel up to 16 conversions may be done per interrupt.

bit 0 ALTS: Alternate Input Sample Mode Select bit

1 = Causes the module to alternately sample between two sets (MUX A and MUX B) of inputs specified by the ADCHS register. On the first sample convert sequence MUX A inputs are selected for sampling and on the second sample and convert sequence MUX B inputs are selected for sampling. This pattern will repeat for subsequent sample conversion sequences.

0 = Only sample input selected by MUX A

For single channel A/D conversion ADCON2 value is,

ADCON2 = 0x0000 or 0b0000000000000000; AV_{DD} and AV_{SS} are the voltage references, no scanning of multiple inputs (single channel), no buffer mode, no interrupts and only sample input selected by MUX A (No alternate input sampling).

1.3 Configuration of A/D control register 3 (ADCON3) is shown in figure 13 and the description of the bits is shown in table 9.

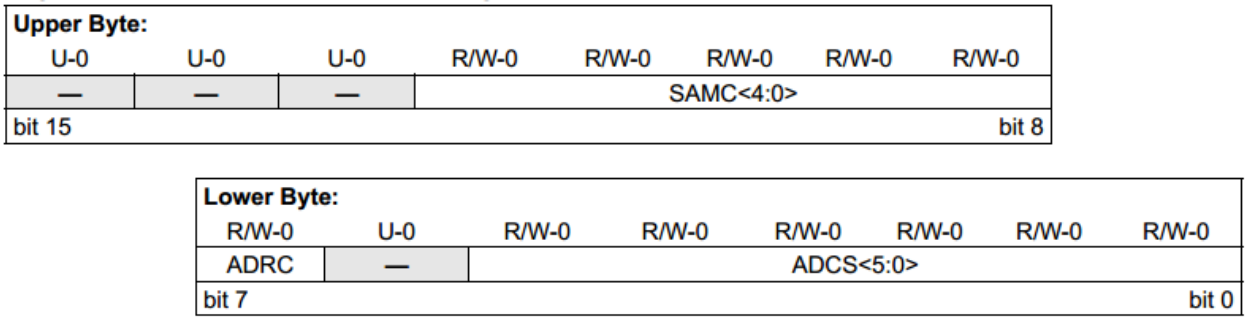


Fig.13. ADCON3: A/D Control Register 3.

Table 9 A/D Control register 3 bits description

ADCON3 bits	Description
bit 15-13	Unimplemented: Read as '0'
bit 12-8	SAMC<4:0>: Auto Sample Time bits, When SSRC<2:0> = 111, the conversion trigger is under A/D clock control. The SAMC bits select the number of TAD clock cycles between the start of sampling and the start of conversion 11111 = 31 TAD ..... 00001 = 1 TAD 00000 = 0 TAD
bit 7	ADRC: A/D Conversion Clock Source bit 1 = A/D internal RC clock, enables the A/D module to operate in sleep mode 0 = Clock derived from system clock T_{cy}
bit 6	Unimplemented: Read as '0'

bit 5-0 ADCS<5:0>: A/D Conversion Clock Select bits

111111 = TCY/2 • (ADCS<5:0> + 1) = 32 • TCY

.....

000001 = TCY/2 • (ADCS<5:0> + 1) = TCY

000000 = TCY/2 • (ADCS<5:0> + 1) = TCY/2

A/D Conversion Clock Calculation

A/D Conversion requires 14 T_{AD}

A/D Conversion Clock $T_{AD} = T_{CY} \times (0.5 \times (ADCS < 5:0 > +1))$

F_{CY} = (Fosc/4) = 20MHz,

T_{CY} = 50nsec

Minimum T_{AD} = 667nsec for correct A/D conversions

ADCS<5:0> = $2 \cdot \frac{T_{AD}}{T_{CY}} - 1$

$= 2 \cdot \frac{667 \text{ nsec}}{50 \text{ nsec}} - 1$

$= 25.68 \cong 26$ (To be on a safer side we took it to be 27)

Actual $T_{AD} = \frac{T_{CY}}{2} \times (ADCS < 5:0 > +1)$

$= \frac{50\text{nsec}}{2} \times (27 + 1)$ or 14 T_{CY}

= 700nsec

Hence, $ADCS < 5:0 >$ = 0b11011, 14 T_{CY} or 27

Acquisition Time > 667nsec or T_{AD} = 700nsec

For T_{AD} based conversion, to select the number of TAD clock cycles between the start of sampling and the start of conversion

SAMC<4:0> = 00010, 2Tad is the conversion trigger source (time between the start of sampling and the start of conversion).

For single channel A/D conversion the value of ADCON3 is,

ADCON3 = 0x021B or 0b0000001000011011; //sample time =2Tad, Tad =internal 14Tcy and A/D conversion clock derived fromT_{CY}.

1.4 A/D Input Select Register (ADCHS): S/H amplifiers are connected to analog input pins via the analog input multiplexer. The ADCHS register selects the input pins to be connected to the S/H amplifiers and also controls the input multiplexer by allowing two different analog input multiplexer configurations (MUXA and MUXB) to be available, by programming the CH0NA & CH0SA <3:0> and CH0NB & CH0SB<3:0> bits corresponding to MUXA and MUXB respectively. The multiplexer of each A/D converter can optionally switch between the MUX A and MUX B configurations between conversions by setting the ALTS bit in the ADCON2 register.

The Sample-and-Hold Amplifier has analog multiplexers on both its inverting and non-inverting terminal to select analog input (and its corresponding reference voltage AV_{SS} or AN1 (external reference voltage$V_{REF\ -}$)) to be sampled. The ADCHS bits perform the same operation for MUX A (CH0SA<3:0> determine the Analog input channel and CH0NB selects the reference) and MUX B (similar for Mux B). Configuration of A/D Input Select Register (ADCHS) is shown in figure 14 and the description of the bits is shown in table 10.

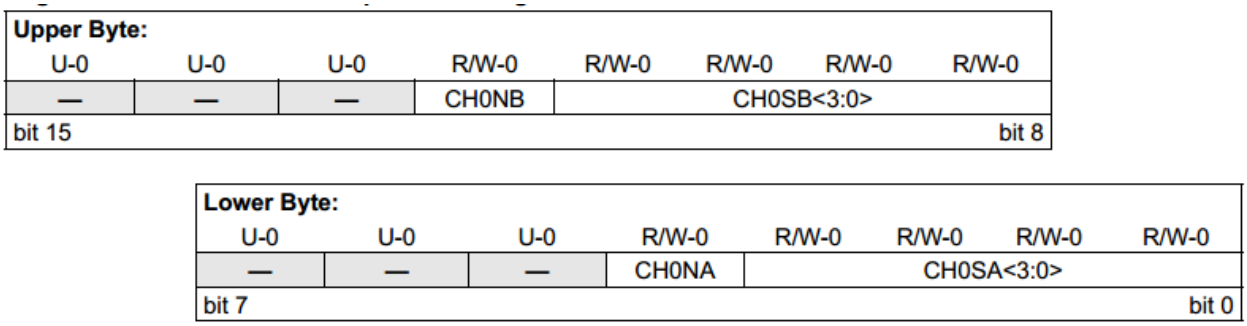


Fig.14. ADCHS: A/D Input Select Register.

Table 10 A/D Input select register bits description

ADCHS bits	Description
bit 15-13	Unimplemented: Read as '0'
bit 12	CH0NB: Channel 0 Negative Input Select for MUX B Multiplexer Setting bit 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is VREF-
bit 11-8	CH0SB<11:8>: Channel 0 Positive Input Select for MUX B Multiplexer Setting bit 1111 = Channel 0 positive input is AN15 1110 = Channel 0 positive input is AN14 1101 = Channel 0 positive input is AN13 ..... 0001 = Channel 0 positive input is AN1 0000 = Channel 0 positive input is AN0
bit 7-5	Unimplemented: Read as '0'
bit 4	CH0NA: Channel 0 Negative Input Select for MUX A Multiplexer Setting bit

1 = Channel 0 negative input is AN1

0 = Channel 0 negative input is VREF-

bit 3-0 CH0SA<3:0>: Channel 0 Positive Input Select for MUX A Multiplexer Setting bit

1111 = Channel 0 positive input is AN15

1110 = Channel 0 positive input is AN14

1101 = Channel 0 positive input is AN13

.....

0001 = Channel 0 positive input is AN1

0000 = Channel 0 positive input is AN0

For single channel A/D conversion

ADCHS = 0x000C; //Connect RB12/AN12 as CH0 input and negative input to the MUX A is VREF-

1.5 A/D Port Configuration Register (ADPCFG) configures analog input channels for the A/D converter module which samples the pin voltage. Configuration of A/D port configuration register (ADPCFG) is shown in figure 15 and the description of the bits is shown in table 11.

Upper Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Fig.15. ADPCFG: A/D Port Configuration Register

Table 11 A/D port configuration register bits description

ADPCFG bits	Description
bit 15-0	PCFG<15:0>: Analog Input Pin Configuration Control bits 1 = Respective PCFG pins are configured as Digital inputs/outputs and the input to the analog multiplexer is connected to AVSS 0 = Respective PCFG pins are configured as Analog Inputs (At reset all the pins multiplexed with AN channels are configured as analog inputs), To properly work as analog inputs there corresponding TRIS bits need to be set.

For single channel A/D converter:

ADPCFG = 0xEFFF, all PORTB = Digital, RB12 = analog;

1.6 ADCSSL: A/D Input Scan Select Register

ADCSSL register specifies the inputs to be sequentially scanned. Each bit in the register corresponds to the analog input for examples bit 0 (CSSL0) corresponds to AN0. Scanning begins from lower to the higher number inputs, for example if we want to scan channels AN1, AN3, AN12, AN5 consecutively, the scanning will take place in the following order AN1, AN3, AN5 and AN12. After every interrupt this order will be followed. Configuration of A/D Input Scan Select register (ADCSSL) is shown in figure 16 and the description of the bits is shown in table 12.

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

Fig.16. ADCSSL: A/D Input Scan Select Register

Table 12: A/D Input Scan Select register bits description

ADCSSL bit	Description
bit 15-0	CSSL<15:0>: A/D Input Pin Scan Selection bits 1 = Select ANx for input scan 0 = Skip ANx for input scan

For single channel A/D converter:

ADCSSL = 0x0000, no scanning

2. We start the conversion by setting the SAMP bit in the ADCON1 register (ADCON1bits.SAMP =1). When SSRC<2:0> = 111, the conversion trigger is under A/D clock control. By configuring the SAMC bits we wait for 2TAD clock cycles between the start of sampling and the start of conversion.

3. Wait for A/D conversion to complete by checking the DONE Bit in the ADCON1 control register. We use while (! ADCON1bits.DONE) to check whether the conversion is completed.

4. Read the result from registers ADCBUF register. We read the result of the A/D conversion from the 16-bit ADCBUF0 buffer register.
5. For next conversion go to Step 2.

2.2.4 Digital Filters: We are using digital filters to extract our frequency band of interest. We have designed analog filters in the hardware section they don't guarantee a prolonged good signal due to aging and external interferences. Once the analog filter parameters are set in the hardware circuit they are difficult to change and also the design on higher order analog filters is very complex. To overcome the above problems we use digital filtering (in addition to analog filters) in our software. Digital filters are basically signal conditioners; i.e. they transmit or reject a frequency range of the original input signal. Digital filter takes digital input from ADC performs mathematical manipulation using digital delays, multiplier and adders to realize a digital filter with far superior level of performance easily (digital filters can implement very high order filters just by simple addition and multiplication).

There are two main types of digital filters they are Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters.

In the FIR filters the output of the filter depends only on previous inputs (FIR filters contain only zeros). To implement FIR filter in a PIC microcontroller only a filtering function of the FIR filter equation needs to be added.

The Basic FIR filter equation is given below

$$y(n) = h_0 \cdot x(n) + h_1 \cdot x(n-1) + h_2 \cdot x(n-2) + .. + h_{N-1} \cdot x(n-(N-1))$$

which can be represented as:

$$y(n) = \sum_{k=0}^{k=N-1} h_k \cdot x(n-k)$$

where $y(n)$is the output of the digital filter at sampling instant n

$x(n)$is the input of the digital filter at sampling instant n

$x(n-1)$is the input of the digital filter at sampling instant $n-1$, or more generally

$x(n-k)$is the input of the digital filter at sampling instant $n-k$

$h_0, h_1 \ldots h_{N-1}$, are the filter coefficients (constants) specific for the particular

N, is the length of the Digital FIR filter

$M = N-1$, is the order of the digital filter

Thus the FIR filter simply produces a weighted average of its N recent input samples.

The longer the filter (more coefficients) the more finely its response can be tuned but the computational load for implementing such a filter increases rapidly and may limit the length of the filter if a relatively high sampling frequency needs to be realized

In case of IIR filters, the output of the IIR filters depends on both previous inputs and outputs (IIR filters consist of zeros and poles).

The Basic IIR filter equation is given below

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + \ldots + b_{N-1} \cdot x\big(n-(N-1)\big) - a_1 \cdot y(n-1) - a_2\, y(n-2) - \ldots - b_M \cdot y(n-M).$$

which can be represented as:

$$y(n) = \sum_{k=0}^{k=N-1} b_k \cdot x(n-k) - \sum_{k=1}^{k=M} a_k \cdot y(n-k)$$

where $y(n)$is the output of the digital filter at sampling instant n

$y(n-1), y(n-2), \ldots y(n-M)$, are M old filter outputs

$x(n)$is the input of the digital filter at sampling instant n

$x(n-1), x(n-2), \ldots, x(n-N+1)$, are $N-1$ old filter inputs

b_0, b_1 ... b_{N-1} and a_0, a_1 ... a_M are two sets of IIR filter coefficients

Coefficients a_0, a_1 ... $a_M$multiply old filter outputs that are fed back into the digital filter, so those coefficients are also sometimes called 'feedback filter coefficients'. Coefficients b_0, b_1 ... $b_{N-1}$are known as 'feedforward filter coefficients'.

Advantages of FIR filters over IIR

1. FIR's are linear phase filters (that is, they delay the input signal but don't distort its phase)and are stable. However IIR have no particular phase relationship and are unstable.
2. FIR have much better delay characteristic than IIR and much simpler to implement. On most microcontrollers, the FIR calculation can be done by looping a single instruction.
3. FIR filters require less number of multiplications and additions for the filter implementation as compared to IIR.
4. Computation speed of FIR is faster than that of IIR.

Disadvantages of FIR over IIR

1. Compared to IIR filters, they require more memory and/or calculation to achieve a given filter response characteristic.

DsPIC30F4013's CPU has a DSP Engine capable of executing Digital FIR and IIR filters with great efficiency and reliability. Microchip provides dsPIC Language Tools Libraries with functions to perform digital filtering alongside other DSP, peripheral and math functions. DSP

library is written in assembly to increase the execution speed and most of its operations are computed using fractional arithmetic to take advantage of the DSP instruction set and architecture and maintain accuracy. As discussed in the ADC section signed fractional format is represented as "sddd dddd dddd 0000" where s is the signed bit and other 15 are the fractional bits (also referred to as "1.15" data format). The input to the digital filter can be in any format (signed/unsigned integer or unsigned fractional), as the DSP engine in the microcontroller automatically converts data into the correct (signed fractional) format. However the output from the digital filter will be in signed fractional format and needs to be converted by the user into the desired format. Microchip also provides digital filter design tool (dsPIC FD) to generate coefficients for desired digital filter response (the Filter Design for dsPIC™ DSC Digital Filter Design and Analysis System manual discusses in detail the usage of the dsPIC FD software) and a filter routine depending on the filter type (FIR, IIR Transpose, IIR Canonic or IIR Canonic Extended Precision).

Note: The filter coefficients generated by the dsPIC FD (can be stored as a filename.s file (assembly file)) and its associated filter routine should be added in the MPLAB project.

DsPIC30F4013's digital filtering is mainly used here to isolate the beta-band (13-28 Hz) and (12-14Hz) and 12-14 Hz band (SSVEP of 13Hz signal) for further processing. It also eliminates the other remaining artifacts from the hardware section (such as 60Hz power line interference, EMG artifacts which overlap with upper and middle parts of the EEG frequency range and eye artifacts (eye blinks) which are predominant in 1-2Hz of EEG frequency range).

We have used FIR Kaiser Window design (gain =1, hence 0-5V represented as 0-4096(12-bit ADC)) to realize our band pass filters of 13-28Hz and 12-14Hz for beta band and SSVEP respectively.

Steps to Configure the Digital Filtering module are as follows:

1. Using the dsPIC FD tool from Microchip we realized our band-pass filters and generated the filter coefficients.
2. We saved our coefficients in the X data space.
3. The 12-bit signed fractional result of the A/D conversion was the input value to the FIR filter function, which performs the filtering and outputs the filtered result. FIR (intnumSamps, fractional* dstSamps, fractional* srcSamps, FIRStruct* filter)

   Where numSamps = 1 (for single sample filtering),

   dstSamps is the pointer to the output value,

   srcSamps is the pointer to the input value,

   FIRStruct is the pointer to the filter coefficients and its associated routine.
4. The filtered output value (in the signed fractional format) was converted to unsigned integer format by right shifting the output value and adding 2049.
5. To get the filtered output of the next A/D result, go to step 3.

2.2.5 Power Calculation

Average Power:

A FIFO (first-in first-out) buffer of N (preset based on the method you are using) bytes is used to store the results (voltage amplitude) from the digital filtering. The average power of which is calculated as follows.

$$P_{avg} = \frac{1}{N} \times \sum_{i=0}^{N} \text{abs}\,(V[i])$$

2.2.6 Experimental paradigm:

EEG signal from a healthy subject was recorded using 4 tin surfaces electrodes (C3 (for beta-rebound based method), OZ (for SSVEP based method), FZ and FZA) attached on an elastic cap (Electro-Cap International, Inc., Eaton, OH, U.S.A.) according to the international 10-20 system[52]. A subject was asked to still comfortably and avoided any motion during the experiment.

A subject performed five sessions of experiment for each method (that is Beta-rebound based method and SSVEP based method). The first session was based on adjusting the power threshold value for the subject. In the remaining session's real time analysis of the brain-controlled switch device was performed. The second session the false positive rate of the BCI switch (the number of times the switch turns ON in the "No control state") was recorded by the investigator. In the next three sessions, the subject was asked to successfully turn on an external switch 10 times per session hence we tested the sensitivity of the brain-controlled switch.

2.2.6.1 Beta-Rebound Based Prediction algorithm: A brisk hand movement (or imagination of the movement) is followed by an increase in amplitude oscillations in the beta band which is also termed as beta rebound. In this method, we focused on the amplitude increase in the beta band associated with the intention of movement to control our external switch.

Considering that subject might be easily fatigued from the sustained active mental task, the subject were able to alternate between a short active task ("Intentional Control" state) and relaxation ("No control state") in the event-related design so that the required period of sustained attention could be greatly reduced which is shown in Fig. 17.

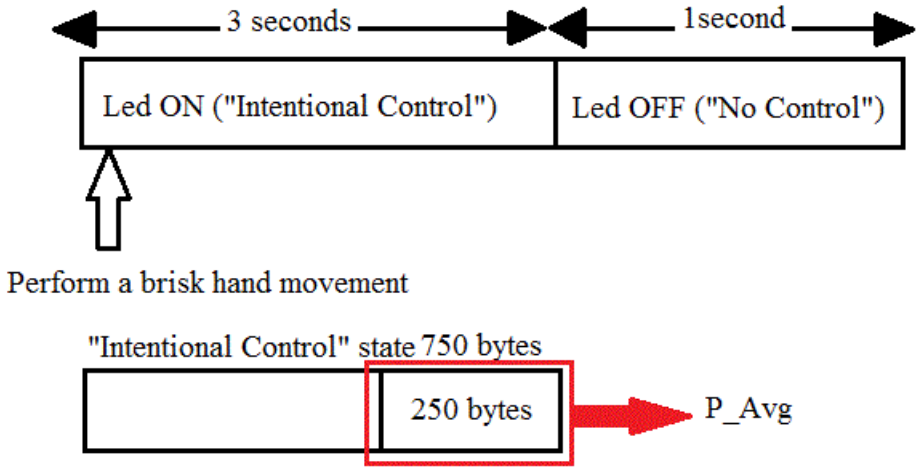


Fig.17. Prediction Algorithm based on Beta Rebound Based Method

From fig.17 it is seen that when the experiment began the subject was provided with an external synchronization signal of 4 seconds (led 'ON' and led 'OFF'). The entire time interval of 4 seconds constituted one trial. Every 4 seconds, LED ON indicated a cue to perform brisk hand movement (or motor imagery based movement). The data for estimation of amplitude power was acquired from the interval of 3 seconds (or "Intentional control" state) where the data length was of 1 second (250 bytes, as the beta rebound is assumed to last for about 1 seconds). Every 4 seconds, average amplitude power was calculated using this 1 second of data. This calculated average amplitude power was compared with the preset power threshold. If the average amplitude power was greater than the power threshold than the external switch was turned ON.

Offline analysis: First session in this experiment was conducted to determine the power threshold for the subject. This session began from 'No control' state, where the subject was instructed to avoid either physical (or imagined hand movement, basically the subject wasn't attentive to the switch in this case). This digitized data was recorded by using the BCI2VR toolbox. Next, the

subject was asked to perform a brisk hand movement in synchronization with external sync signal ("Intentional control" state). This data was also recorded. We calculated the offline average power between the "Intentional control" state and the "No control" state. And based on this comparison, power threshold value was loaded in our microcontroller.

Performance evaluation: Real time analysis of our Brain-controlled switch device based on the beta-rebound method was performed. A total of four sessions were performed here. In the first session the subject's "No control state" operation was monitored by the investigator for 3 minutes. In the 'No control state', the subject was instructed to carry out normal thinking and the investigator noted the number of false positive detections (that is, the number of times the led turned ON in the "No Control" state) made. In the next three sessions, the subject was asked to successfully turn ON the external switch 10 times. These sessions tested the sensitivity of our device. Here the investigator recorded the time it took to successfully turn ON the external switch each time.

2.2.6.2 SSVEP based prediction algorithm:: In this method, when the user concentrates on the flickering LED, a dominant fundamental frequency (equivalent to the flickering frequency) appears in the spectral representation of the EEG signals recorded at occipital lobes (location OZ). Here, we focus on the amplitude increase in the band containing the fundamentally frequency which is used to control the external switch.

In this experiment, when the user decides to operate the external switch he/she focuses his/her attention onto the LED flashing at 13Hz which is placed at a distance of 1 meter in front of the subject this marks the "Intentional Control" state. In the "No control state" the subject looks away from the flashing LED. Every 0.5 seconds, our microcontroller calculates the average

amplitude power of the data recorded in the previous 0.8 seconds as shown in Fig. 18. The obtained average amplitude power was compared with the preset power threshold. If the average amplitude power was greater than the power threshold the external switch is turned ON.

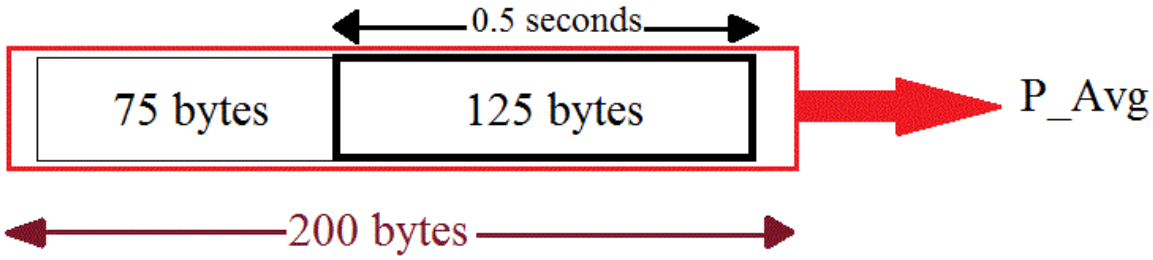


Fig.18. Prediction Algorithm Based on SSVEP Based Method

Offline analysis: Similar to beta rebound based method, the first session in this experiment was conducted to determine the power threshold for the subject. This session began from 'No control' state, where the subject was instructed to look away from the flashing LED. This digitized data was recorded by using the BCI2VR toolbox. Next, the subject was asked to concentrate on the flashing LED ("Intentional control" state). This data was also recorded. We calculated the offline average power between the "Intentional control" state and the "No control" state. And based on this comparison, power threshold value was loaded in our microcontroller.

Performance evaluation: Real time analysis of our Brain-controlled switch device based on the SSVEP method was performed. A total of four sessions were performed here. In the first session the subject's "No control state" operation was monitored by the investigator for 3 minutes. In the 'No control state', the subject was instructed look away from the LED and the investigator noted the number of false positive detections (that is, the number of times the led turned ON in the "No Control" state) made. In the next three sessions, the subject was asked to successfully turn ON

the external switch 10 times. These sessions tested the sensitivity of our device. Here the investigator recorded the time it took to successfully turn ON the external switch each time.

2.2.7 Algorithms for the design

Algorithm for beta-rebound based method:

1. Every 4msec (sampling rate of 250Hz) from the Timer1.
    i. The amplified EEG signal from the hardware section is converted into a digital signal by the A/D Converter.
    ii. This signal is filtered to obtain a band-pass signal of 13-28Hz (beta band for the beta-rebound based calculations).
    iii. And the output of the digitized filtered signal is stored in the buffer V [250 Bytes].
2. Timer 23 is used here to provide an external synchronization signal (3 second ON and 1 second OFF). The ON period indicates the "Intentional Control" state and the OFF period indicates the "No Control" state. Every 4 seconds, average power of the buffer V [250 bytes, 1 second of data] is calculated. The average amplitude power is compared to the preset power threshold (Power at rest). If the average power is greater than the power threshold than the LED is turned ON.

Algorithm for SSVEP based method:

1. Every 4msec (sampling rate of 250Hz) from the Timer1.
    i. The amplified EEG signal from the hardware section is converted into a digital signal by the A/D Converter.
    ii. This signal is filtered to obtain a SSVEP band of 12-14Hz containing the 13Hz SSVEP fundamental frequency.

iii. And the output of the digitized filtered signal is stored in the buffer V[200 Bytes].

2. Timer 23 is used here to calculate the power of the buffer V [200 bytes] every 0.5 second. The average power is compared to the power threshold, and if the average power is greater than the threshold the LED is turned ON.

CHAPTER 3

RESULTS

3.1 BCS device's signal recording module's evaluation: First the brain controlled switch device was used to capture the ECG signal using three electrodes connected to the right wrist, left wrist and right leg of the subject. A successful QRS ECG signal was measured.

Before every experiment, calibration of the Brain-controlled switch system was performed. The basic task of the calibrator was to set overall sensitivity of the Brain-controlled interface system. That is, to establish a known and fixed correspondence between the EEG signals appearing at the inputs and outputs (numbers) that would emerge from the A/D converters of dsPIC30F4013 at the computer interface. EEG Calibrator (Modina Bio-Engineering Version 1.3 2) was used. It is a sine wave generator that can produce 1 to 64Hz sine waves of known, fixed amplitude (50µV, 20 µV and 2000 µV). We used the calibrator to generate a 50 µV sine wave at 20Hz which was used to test our BCS system. The output from the brain controlled switch was adjusted to set the offset to 0 and the gain to unity.

Subject preparation: As mentioned earlier an EEG signal from a healthy subject was recorded using 4 tin surfaces electrodes (C3 (for beta rebound based method), OZ (for the SSVEP based method), FZ (reference) and FZA (DRL)) attached on an elastic cap (Electro-Cap International, Inc., Eaton, OH, U.S.A.) according to the international 10-20 system[52]. Skin artifacts were reduced using skin cleansing solution to wet and clean the skin surface to remove debris, oils, and damaged or dead epidermal cells and electrode gel was applied to maintain a high-quality interface between the electrode metal and the skin.

Bio-potential amplifier Evaluation: Initially a pure and amplified EEG signal was recorded from the BCS device (using just the bio-potential amplifier and the A/D conversion module) at a sampling rate of 250Hz and displayed on the computer screen of the BCI2VR toolbox using the UART2 module. This is shown in Fig. 19:

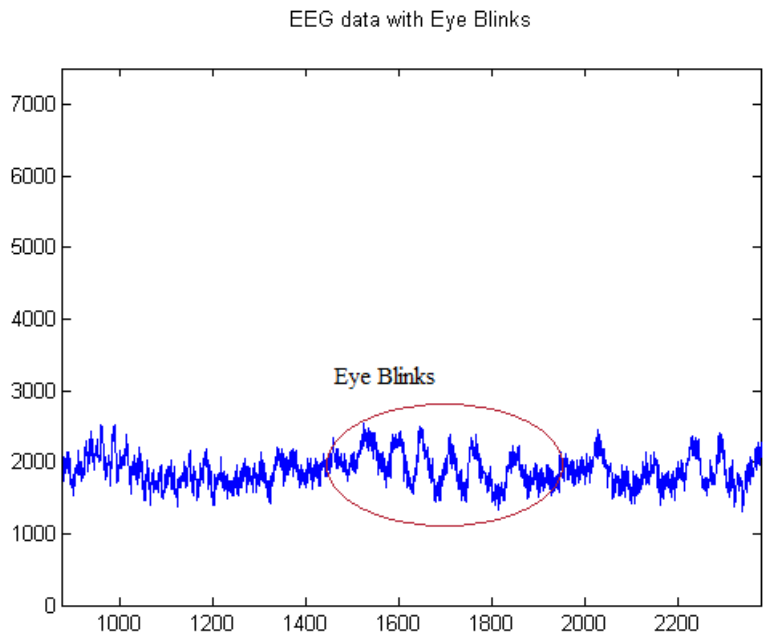


Fig.19. Pure EEG Signal Measured from the Scalp.

The digitized EEG signal was recorded with $\pm$ 0.2 % error by the A/D conversion module. EOG signal induced by the eye-movement and eye blinks was predominant source of artifact at frontal recording sites existing in the 1-8Hz frequency bands. In our device as we filtered out the frequency bands associated with EOG signal, hence it didn't affect our system performance and our subject weren't asked to intentionally reduce eye-movements and blinks.

It was also seen that without the DRL unit of the bio-potential amplifier, our brain-controlled switch device measurement was saturated which is shown in Fig. 20.

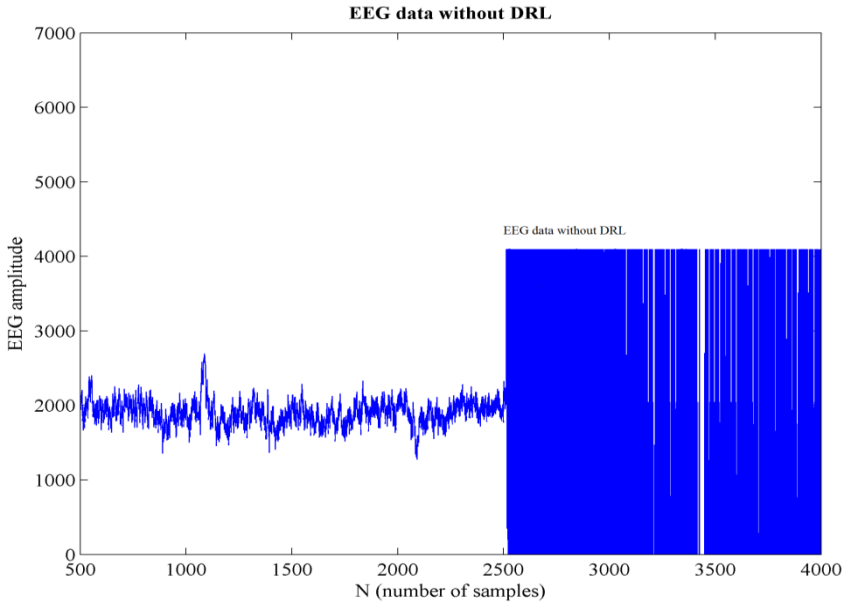


Fig.20. EG Signal Saturation Without the DRL Circuit.

3.2 BCS device's feature extraction module evaluation: our device aimed to extract the frequency bands of 13-28Hz and 12-14Hz for the beta rebound based method and the SSVEP based method with the fundamental frequency of 13 Hz respectively. From bode plot responses shown in Fig.21 and Fig.22, we can see that our device was successful in extracting the desired bands of interests with a unity gain.

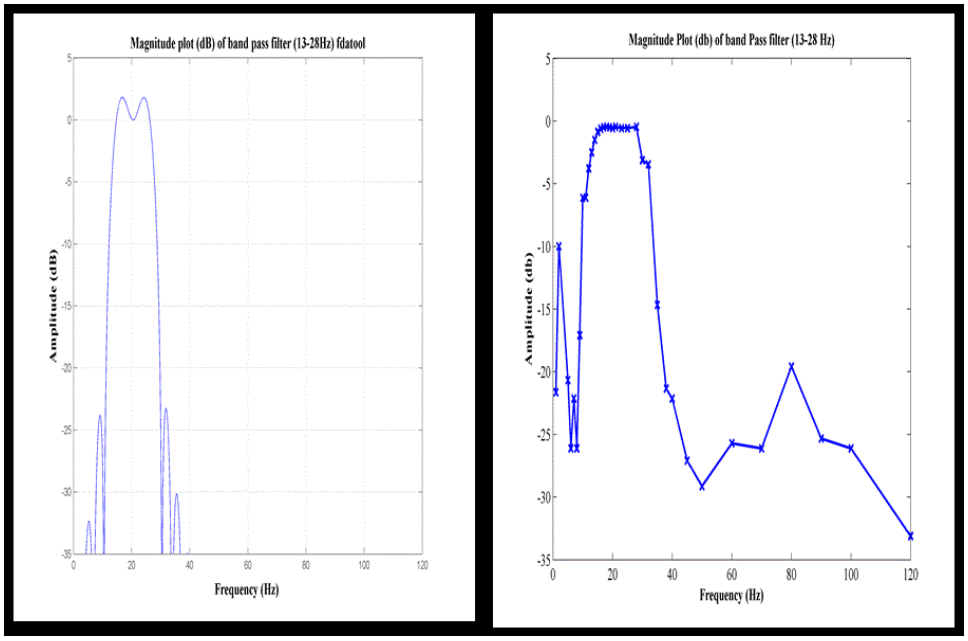


Fig.21. Bode Plot of the Beta Band (13-28Hz) Filter for the Beta-rebound Based Method, FIR Kaiser Window Design with taps =64.

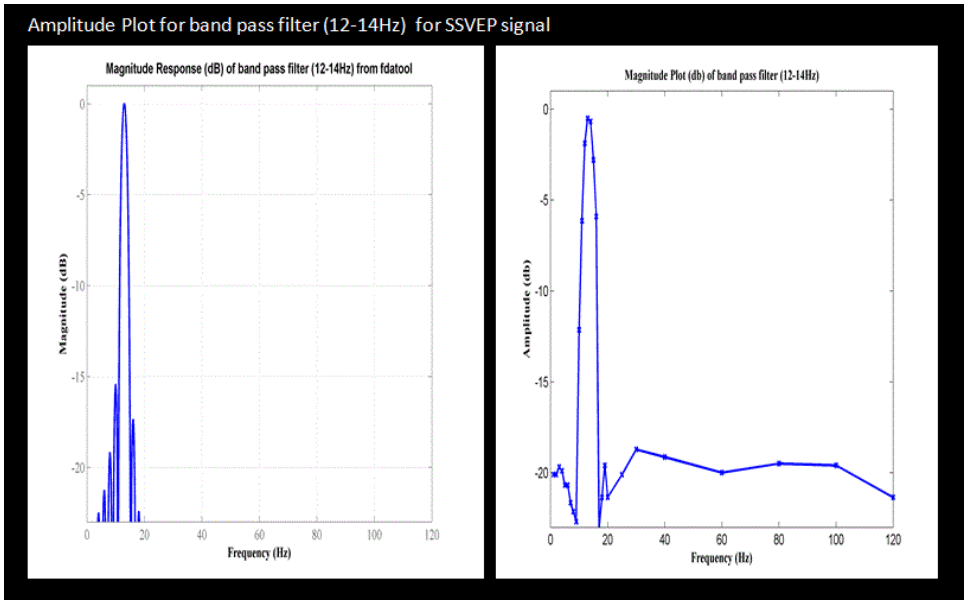


Fig.22. Bode Plot of the Band Pass (12-14Hz) Filter for SSVEP Signal of13Hz, FIR Kaiser Window Design with taps =125.

3.3 BCS performance: Beta-rebound based method and SSVEP based method are stimulus based designs as both rely on the user to produce a specific mental task in response to a specific stimulus. The mental task is thus time-locked to the stimulus. We performed a multi-session study, based on each method to test the reliability and robustness of the brain-controlled switch performance during long-time operation.

3.3.1. BCS performance based on beta-rebound method: In this method, the stimulus is a synchronization signal from the LED of interval 4 sec (3 seconds ON and 1 sec OFF). Every 4 seconds, when the synchronization LED is turn ON (go-signal provided) the user should perform a brisk hand movement if he/she indents to turn ON the external switch, the beta rebound complex resulting from this movement is time locked to the stimulus.

In the experiment based on this method, a subject participated in a five session study, while the subject was operating the switch the computer made detections after every 4 seconds to determine whether the subject intended to switch ON the LED.

The first session was conducted to calculate the power threshold of the subject. This session began with recording of the 'No control' state of the subject. Next the subject was asked to perform a brisk hand movement in synchronization with the go signal provided in 'Intentional control' state and the data was recorded. The ERD/ ERS (beta-rebound) complex generated from this session is shown in Fig 23.

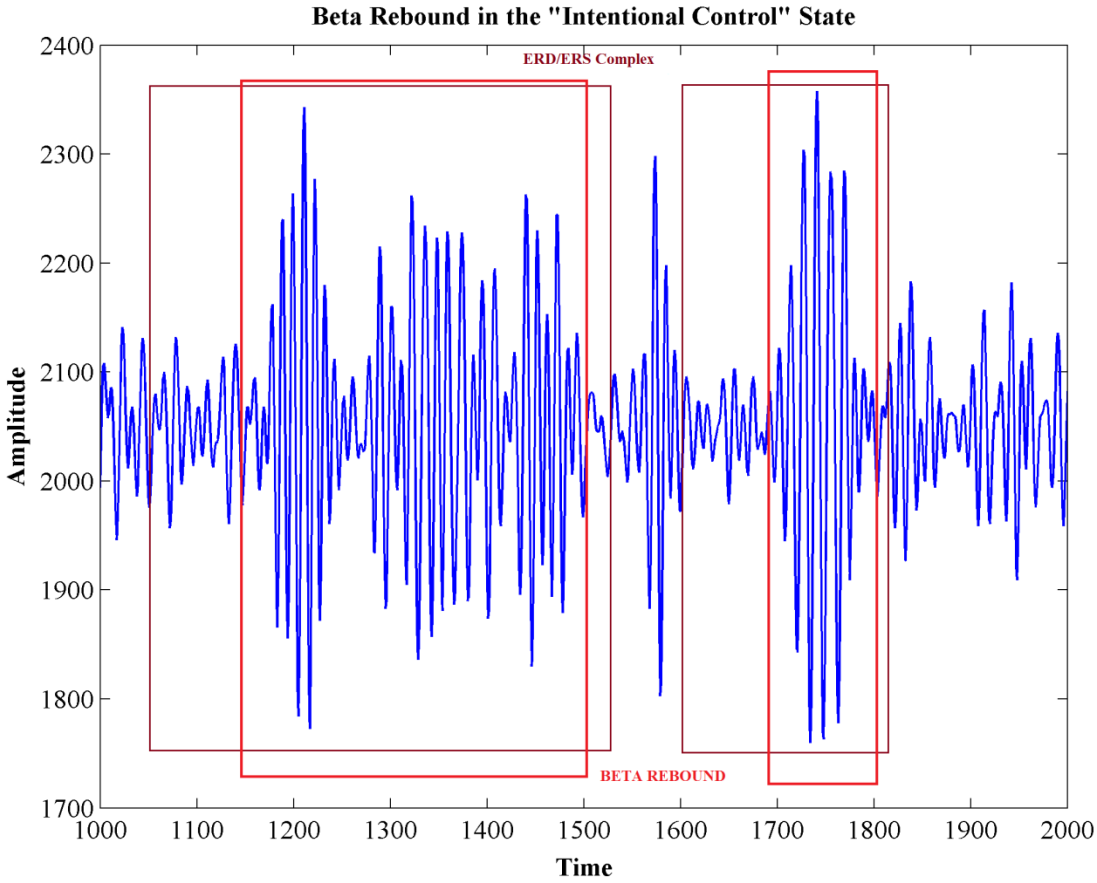


Fig.23. Offline Analysis of the Session Based on Beta-rebound Based Prediction Algorithm

A very ERD/ERS complex was seen in the beta band of the subject. An offline analysis was performed where the average offline amplitude power spectrum of 'Intentional' and 'No control' state for 2.3 minutes of data was calculated based on our beta-rebound based prediction algorithm and is amplitude power spectra is shown in Fig. 24.

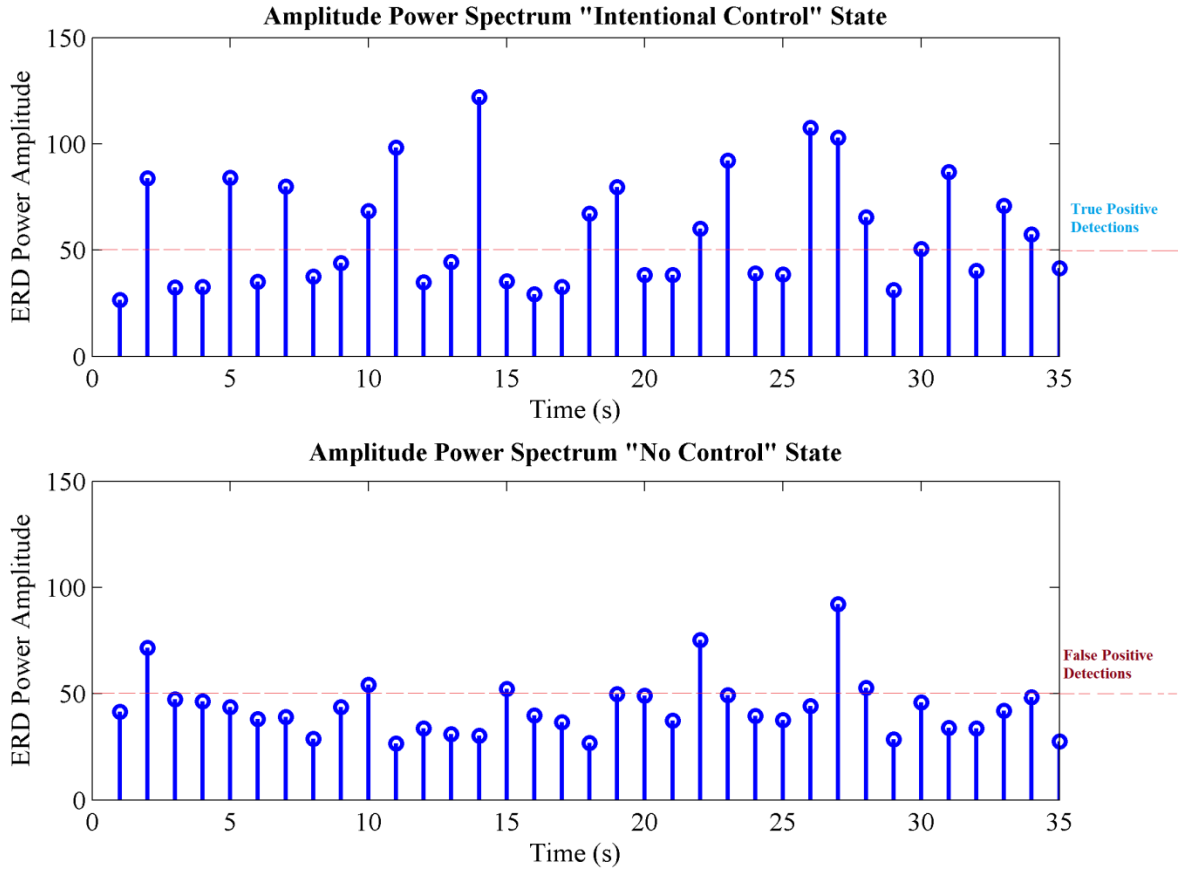


Fig.24. Offline Analysis Amplitude Power Spectra for the "Intentional Control" and "No Control" State

From the analysis of the Fig.24, power threshold value was set to 50 as it had more number of true detections (that is in the "Intentional Control" state the number of times the switch is turned ON when the user performs a brisk hand movement every 4 seconds) and lesser false positive detections (number of times the switch turns ON in the "No-control" state).

False positives: In the next session, an investigator monitored the external LED for the 'No Control' state of the subject and noted the number of times the LED turned ON in 3 minutes. The switch was activated 4 times in this session. That is, there were about 1.3 false detections out of

15 predictions during the “No control state for a single session” and the FPR was the percentage of false detections made from the total detections during the no control state. Thus the false positive rate was 9.9%.

Sensitivity: In the next three sessions, the subject was asked to successfully turn ON the external switch 10 times by performing a brisk hand movement in synchronization with the external sync signal “Intentional control” state. Here the investigator recorded the time it took to successfully turn ON the external switch each time. The average response time required by the subject to turn ON switch successfully per session is shown in table:

Table 12: Performance of the switch based on beta-rebound method in three sessions

No. of session	Mean (seconds)	Standard deviation (seconds)
Session 1	12.62	5.3167
Session 2	11.65	8.3559
Session 3	12.51	8.8097
Total	12.23	7.39843

Thus in the intentional control stage overall urging time to activate the switch for the subject was about 12.23± 7.39 seconds. Thus per minute the switch was turned on 5 times correctly per minute.

3.3.2 BCS performance based on SSVEP method: In this method, the stimulus is a flashing or flickering LED of 13Hz frequency. When the user intents to switch ON the external LED he/she should concentrate on the flashing LED, thus this method is time-locked with the stimulus.

In the experiment based on this method, a subject participated in a five session study, while the subject was operating the switch the computer made detections after every 0.5 seconds to determine whether the subject intended to switch ON the LED.

The first session was conducted to calculate the power threshold of the subject. This session began with recording of the 'No control' state of the subject where the subject looked away from the flashing LED. Next the subject was asked to concentrate on the flashing LED and the data was recorded "Intentional control" state. The offline analysis of the session based on SSVEP based prediction algorithm is shown in Fig. 25 which clearly shows an increase in amplitude in the "Intentional control state" when compared to the "No Control" state.

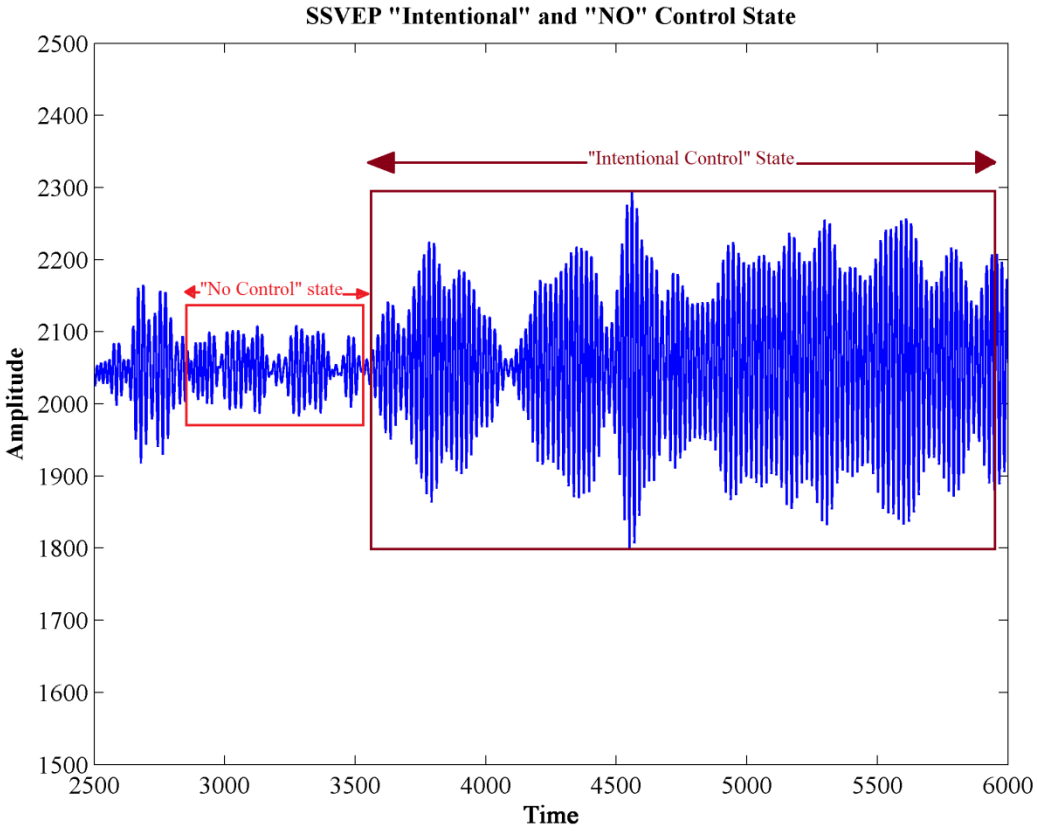


Fig. 25. Offline Analysis of the Session Based on SSVEP Based Prediction Algorithm.

The amplitude average power was calculated offline based-on the SSVEP based prediction algorithm which is shown in the Fig.26.

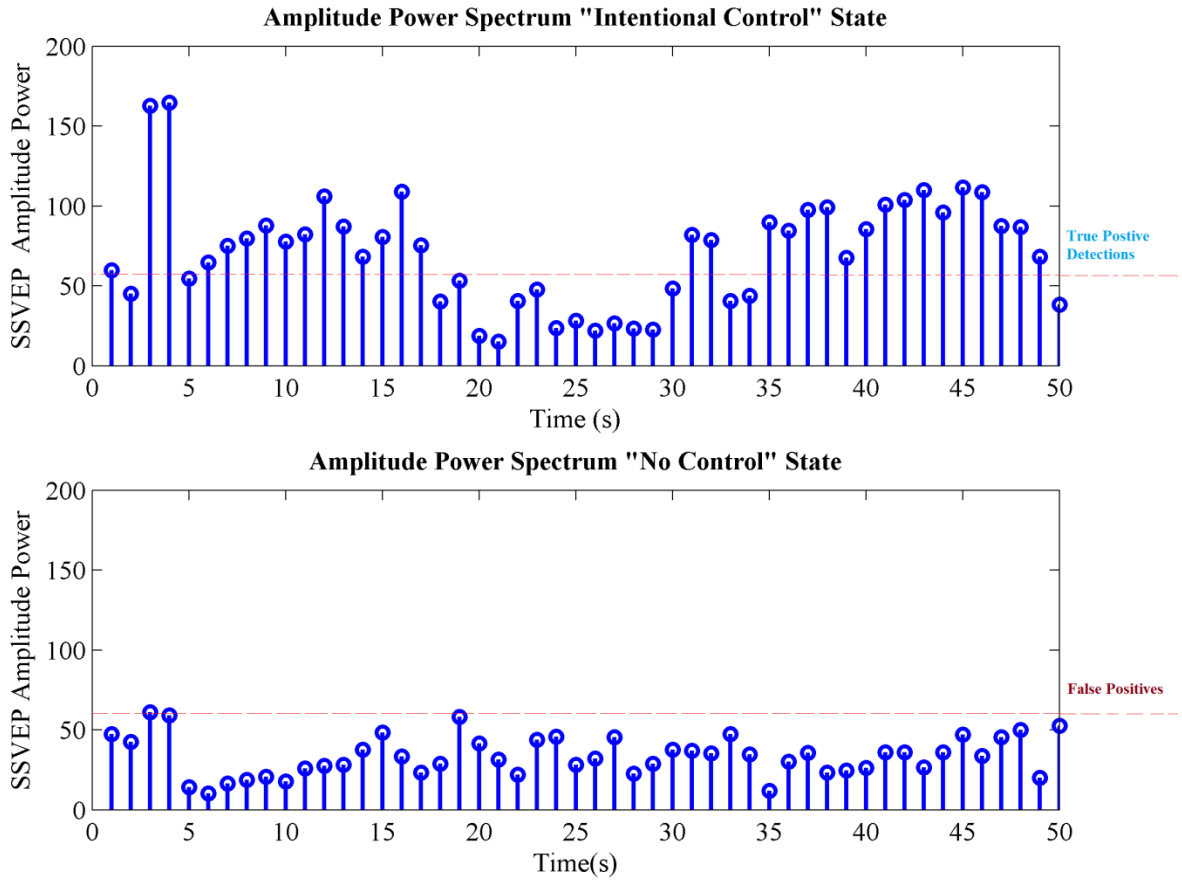


Fig.24. Offline Analysis of the Amplitude Power Spectrum for the "Intentional Control" and "No Control" State Based on the SSVEP Method.

From the analysis of the above figure, power threshold value was set to 60 as it provided maximum true detections and minimum false positives rates.

False positives: In the next session, an investigator monitored the external LED for the 'No Control' state of the subject and noted the number of times the LED turned ON in 5 minutes. The switch was activated 7 times in this session. That is, there were about 1.4 false detections out of 120 predictions during the "No control state for a single session" and the FPR was the percentage

of false detections made from the total detections during the no control state. Hence the false positive rate of 1.17%.

Sensitivity: In the next three sessions, the subject was asked to successfully turn ON the external switch 10 times by concentration on the flashing LED "Intentional control" state. Here the investigator recorded the time it took to successfully turn ON the external switch each time. The average response time required by the subject to turn ON switch successfully per session is shown in table 13.

Table 13: Performance of the switch based on SSVEP based method in three sessions

No. of trails	Mean (seconds)	Standard Deviation (seconds)
Session 1	2.62	0.7204
Session 2	3.02	0.9217
Session 3	3.79	1.9123
Total	3.14	1.3442

Thus in the intentional control stage overall urging time to activate the switch for the subject was about 3.14 ± 1.3442 seconds. Thus overall the switch activates 19 times per minute in the "Intentional control" state.

Therefore, this method provides a potential means to distinguish between "Intentional Control" and "No control" state.

Chapter 4

DISCUSSION AND FUTURE WORK

4.1 Discussion

The performance a simple real-world ON/OFF switch is mainly determined by the sensitivity (how difficult it is to activate the switch), and the specificity (switch is activated only when we intend to activate it). There is always a tradeoff between sensitivity and specificity. For example we can design the system with low sensitivity so that the users need to exert more effort to activate the switch thereby increasing the specificity (that is decreasing the possibility of miss operation). Also in case of real-world switch the hand operation is very reliable. To design a brain controlled switch with similar reliability is challenging because the brain signals have a very low signal to noise ratio. We have tried to design a brain controlled switch equivalent to the real world low sensitivity switch, which allows the user to activate the switch only when they intend to do so by introducing the "Intentional control" state phenomenon.

In this study we have designed and implemented a "stand-alone BCS device", which could be used as a tool to evaluate the performance of the device based on ERS (beta rebound) feature associated with human's natural motor control (movement or motor imagery) as well as SSVEP elicited in human's brain in response to active concentration on the visual stimulus flashing at a certain frequency. Both ERS and SSVEP are physiological processes. However we found a great variability in the ERS patterns for our subject. This variance in ERS caused a high false positive rate (number of times the switch is activated without user's intentions) of 9.3 %. In order to improve the FPR and the sensitivity (average response time required to activate the switch), we

suggest to implement a multichannel EEG system with advance signal processing methods for feature extraction and classification. We would also suggest implementing the difference between ERS and ERDs in the "No control" and "intentional control" state as prediction feature rather than just using ERS, to improve accuracy of the system. In case of SSVEP based design, we achieved an acceptable false positive rate of 1.1% with an average response time 3.1 ± 1.3 seconds for the subject. However the subject complained of fatigue and annoyance caused by concentrating on the LED for a prolonged time interval. We suggest substituting our visual stimulus (flashing LED) by Graphic Stimulus or pattern reversal methods and then test system's performance.

In both beta rebound and SSVEP based methods the power threshold was set manually, thus optimization of power threshold should also be considered. We also suggest including more number of normal and target subjects for experiments to better evaluate the performance of our BCS device. We expect that the proposed brain-controlled switch device can serve as a plausible real-world switch that allows its users to operate it, by only using their mind.

Applications of the Current Brain-Controlled Switch system:

1. Locked-in patients could answer Yes/No questions using this device (LED ON equivalent to Yes and LED OFF to No).
2. By replacing the LED with the buzzer, bed ridden patients could notify their nurses that they require attention.
3. Also this device could be used in conjunction with software that displays an alphabet on screen every four milliseconds. If user intents to select the alphabet he/she should perform motor imagery task that lights the LED (or the microcontroller could transfer data (of user intent to select the alphabet) to the computer.

4. One dimensional (right and left cursor movement) cursor control using ERS of the left and right motor areas corresponding right and left motor imagery hand movement. Two dimensional cursor control to use ERS and ERD's of the left and right motor areas corresponding right and left motor imagery hand movement to move the cursor (right/left corresponding to the ERS) and (up/down corresponding to the ERDs))[53] .

4.2 Future Work

There are significant improvements that can be made to Brain controlled switch design, in order to improve its performance. Some ideas are summarized below:

1. EEG calibration and Impedance match should be implemented to enable a completely stand-alone Brain controlled switch. Rather than using an external calibrator to calibrate the device before each experiment, we can design a calibrator in our microcontroller which can internally produces a 50µV sine wave at frequencies 20Hz for beta-rebound or 13Hz for SSVEP based method. And a push button can be used to initiate calibration. Any disturbance between electrode and the scalp (such as electrode movement or sweat reaction with metal)can alter the electrode impedance[54], which can interfere with the EEG signal and alter its amplitude. Impedance between each electrode should be a low as possible. Our microcontroller could monitor the electrode impedances throughout the experiment if any change occurs it can display on the LCD.
2. The EEG data from the 12-bit A/D channel was measured with 99.5 % accuracy. It didn't seem to affect the EEG data much. We suggest to implement the system using 16-bit (AD7680) or 24-bit (AD7764) analog to digital converters which are ideal for high speed data acquisition of analog signals to ensure very accurate EEG measurements.

3. Rather than connecting the RS232 module to the computer for EEG data processing. An attempt should be made to look into XBee module (which can be directly connected to the UART ports RX and TX of the dsPIC microcontroller). It seems to be a convenient alternative for the user in terms of mobility.

4. For beta-band rebounds relying on visual stimulus (LED ON for attempt to perform a motor task) poses a problem in the advance stages of locked-in syndrome, as eye muscle paralysis occurs which causes visual impairment[55]. Thus this system can't be beneficial for the locked-in users with visual impairment. An alternative to visual stimulus is auditory stimulus (LED ON and OFF in the system can be replaced by "Da" sound (perform motor imagery task) and "Di" sound (no-control/rest state)), it could be tested and its results compared.

5. A self-powered EEG system could be developed from the body heat using Micro energy scavengers (also called thermoelectric generators which convert thermal energy to electric energy)[56, 57].

Literature Cited

REFERENCES

1. Berger, H., *Uber das Electrenkephalogramm des Menchen.* Arch PsychiatNervenkr, 1929. **87**: p. 527–570.
2. Laureys, S., A.M. Owen, and N.D. Schiff, *Brain function in coma, vegetative state, and related disorders.* Lancet Neurol, 2004. **3**(9): p. 537-46.
3. Albert, S.M., et al., *Wish to die in end-stage ALS.* Neurology, 2005. **65**(1): p. 68-74.
4. Mitsumoto H, N.F., *Amyotrophic Lateral Sclerosis: A Comprehensive Guide to Management.* New York: Demos Publication, 1994.: p. pp. 1 – 20.
5. Birbaumer, N., A.R. Murguialday, and L. Cohen, *Brain-computer interface in paralysis.* Curr Opin Neurol, 2008. **21**(6): p. 634-8.
6. Nnez, P.L., *Electric fields in the brain: The neurophysiocs of EEG.* New Oxford University Press, 1981.
7. Lebedev, M.A. and M.A. Nicolelis, *Brain-machine interfaces: past, present and future.* Trends Neurosci, 2006. **29**(9): p. 536-46.
8. Schmidt, E.M., *Single neuron recording from motor cortex as a possible source of signals for control of external devices.* Ann Biomed Eng, 1980. **8**(4-6): p. 339-49.
9. Wessberg, J., et al., *Real-time prediction of hand trajectory by ensembles of cortical neurons in primates.* Nature, 2000. **408**(6810): p. 361-5.
10. Serruya, M.D.e.a., *Instant neural control of a movement signal.* Nature 2002. **416**: p. 141–142.
11. Taylor, D.M., S.I. Tillery, and A.B. Schwartz, *Direct cortical control of 3D neuroprosthetic devices.* Science, 2002. **296**(5574): p. 1829-32.

12. Carmena, J.M., et al., *Learning to control a brain-machine interface for reaching and grasping by primates.* PLoS Biol, 2003. **1**(2): p. E42.

13. Andersen, R.A., S. Musallam, and B. Pesaran, *Selecting the signals for a brain-machine interface.* Curr Opin Neurobiol, 2004. **14**(6): p. 720-6.

14. Scherberger, H., M.R. Jarvis, and R.A. Andersen, *Cortical local field potential encodes movement intentions in the posterior parietal cortex.* Neuron, 2005. **46**(2): p. 347-54.

15. Rosso, O.A. and M.L. Mairal, *Characterization of time dynamical evolution of electroencephalographic epileptic records.* Physica a-Statistical Mechanics and Its Applications, 2002. **312**(3-4): p. 469-504.

16. Sellers, E.W., T.M. Vaughan, and J.R. Wolpaw, *A brain-computer interface for long-term independent home use.* Amyotroph Lateral Scler, 2010. **11**(5): p. 449-55.

17. Birbaumer, N., et al., *A spelling device for the paralysed.* Nature, 1999. **398**(6725): p. 297-8.

18. Hinterberger, T., et al., *A brain-computer interface (BCI) for the locked-in: comparison of different EEG classifications for the thought translation device.* Clin Neurophysiol, 2003. **114**(3): p. 416-25.

19. J.R. Wolpaw, D.J.M., T.M. Vaughan, *Brain-Computer Interface at the Wadsworth Center*. Ieee, 2000. **8**: p. 222-226.

20. Qian, K., et al., *A motor imagery-based online interactive brain-controlled switch: paradigm development and preliminary test.* Clin Neurophysiol, 2010. **121**(8): p. 1304-13.

21. G. Pfurtscheller, C.N., C. Guger, W. Harkam, H. Ramoser, A. Schlögl, B. Obermaier, and M. Pregenzer, *Current trends in grazbrain-computer interface (BCI) research.* IEEE Trans. Rehab. Eng., 2000. **8**: p. 216–219.

22. Penny, W.D., et al., *EEG-based communication: a pattern recognition approach.* IEEE Trans Rehabil Eng, 2000. **8**(2): p. 214-5.

23. Sutter, E.E., *The brain response interface: Communication through visually-induced electrical brain response.* J. Microcomput. Appl., 1992. **15**: p. 31–45.

24. P. J. Cilliers, V.d.K., and J. W. Andre, *VEP-based computer interface for C2-quadriplegics.* Annu. Conf. Engineering in Medicine and Biology, 1993: p. 12.

25. Wolpaw, J.R., et al., *Brain-computer interfaces for communication and control.* Clin Neurophysiol, 2002. **113**(6): p. 767-91.

26. Birbaumer, N., *Brain-Computer-Interface Research: coming of age.* Clin. Neurophysiology, 2006. **117**: p. 479-483.

27. Birch, G.E., Z. Bozorgzadeh, and S.G. Mason, *Initial on-line evaluations of the LF-ASD brain-computer interface with able-bodied and spinal-cord subjects using imagined voluntary motor potentials.* IEEE Trans Neural Syst Rehabil Eng, 2002. **10**(4): p. 219-24.

28. Bashashati, A., et al., *An improved asynchronous brain interface: making use of the temporal history of the LF-ASD feature vectors.* Journal of Neural Engineering, 2006. **3**(2): p. 87-94.

29. Parini, S., et al., *A robust and self-paced BCI system based on a four class SSVEP paradigm: algorithms and protocols for a high-transfer-rate direct brain communication.* Comput Intell Neurosci, 2009: p. 864564.

30. Faradji, F., R.K. Ward, and G.E. Birch, *Plausibility assessment of a 2-state self-paced mental task-based BCI using the no-control performance analysis.* J Neurosci Methods, 2009. **180**(2): p. 330-9.

31. Fatourechi, M., et al., *Automatic user customization for improving the performance of a self-paced brain interface system.* Med Biol Eng Comput, 2006. **44**(12): p. 1093-104.

32. Jurkiewicz MT, G.W., Bostan AC, Cheyne, *Post-movement beta rebound is generated in motor cortex: evidence from neuromagnetic recordings.* Neuroimage, 2006.

33. Vaughan, T.M., J.R. Wolpaw, and E. Donchin, *EEG-based communication: prospects and problems.* IEEE Trans Rehabil Eng, 1996. **4**(4): p. 425-30.

34. Vidal, J.J., *Toward direct brain-computer communication.* Annu Rev Biophys Bioeng, 1973. **2**: p. 157-80.

35. Xiaorong Gao, D.X., Ming Chang, *A BCI-based environmental controller for the motion disabled.* . IEEE transactions on Neural systems and rehabilitation Engineering, 2003. **11**: p. 137-140.

36. Burke, M.J. and D.T. Gleeson, *A micropower dry-electrode ECG preamplifier.* IEEE Trans Biomed Eng, 2000. **47**(2): p. 155-62.

37. Obeid, I., et al., *Two multichannel integrated circuits for neural recording and signal processing.* IEEE Trans Biomed Eng, 2003. **50**(2): p. 255-8.

38. *http://openeeg.souceforge.net*.

39. Huhta, J.C. and J.G. Webster, *60-HZ interference in electrocardiography.* IEEE Trans Biomed Eng, 1973. **20**(2): p. 91-101.

40. M.Teplan, *Fundamentals of EEG measurements* Measurement science review, 2002. **2**.

41. Zhang Lei, G.X.-j., Wu Xiao-pei, *Design and implementation of a BCI System based onsteady-state visual evoked potentials.* ieee, 2010.

42. Nammi, K., *Voltage-Limiting Devices for ECG amplifiers.* Medical Instrumentation, 2003. **BME 462**.

43. P. Horowitz and W. Hill, *The Art of Electronics.* Cambridge UniversityPress, 1989. **2nd ed**.

44. Lin, Y.D., et al., *Preamplifier with a second-order high-pass filtering characteristic.* IEEE Trans Biomed Eng, 1999. **46**(5): p. 609-12.

45. Winter, B.B. and J.G. Webster, *Driven-right-leg circuit design.* IEEE Trans Biomed Eng, 1983. **30**(1): p. 62-6.

46. Webster, J.G., *Reducing motion artifacts and interference in biopotential recording.* IEEE Trans Biomed Eng, 1984. **31**(12): p. 823-6.

47. Bai, O., et al., *Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG.* Clin Neurophysiol, 2007. **118**(12): p. 2637-55.

48. Bai, O., et al., *A high performance sensorimotor beta rhythm-based brain-computer interface associated with human natural motor behavior.* J Neural Eng, 2008. **5**(1): p. 24-35.

49. <*http://ww1.microchip.com/downloads/en/DeviceDoc/70046D.pdf*>.

50. *http://ww1.microchip.com/downloads/en/devicedoc/70138c.pdf.*

51. *http://ww1.microchip.com/downloads/en/devicedoc/51456b.pdf,*.

52. Jasper, H.H.a.H.L.A., *Electro-encephalography . III. Normal differentiation of occipital and precentral regions in man.* Arch Neurol Psychiat, 1938. **39**: p. 95-115.

53. Huang, D., et al., *Decoding human motor activity from EEG single trials for a discrete two-dimensional cursor control.* J Neural Eng, 2009. **6**(4): p. 046005.

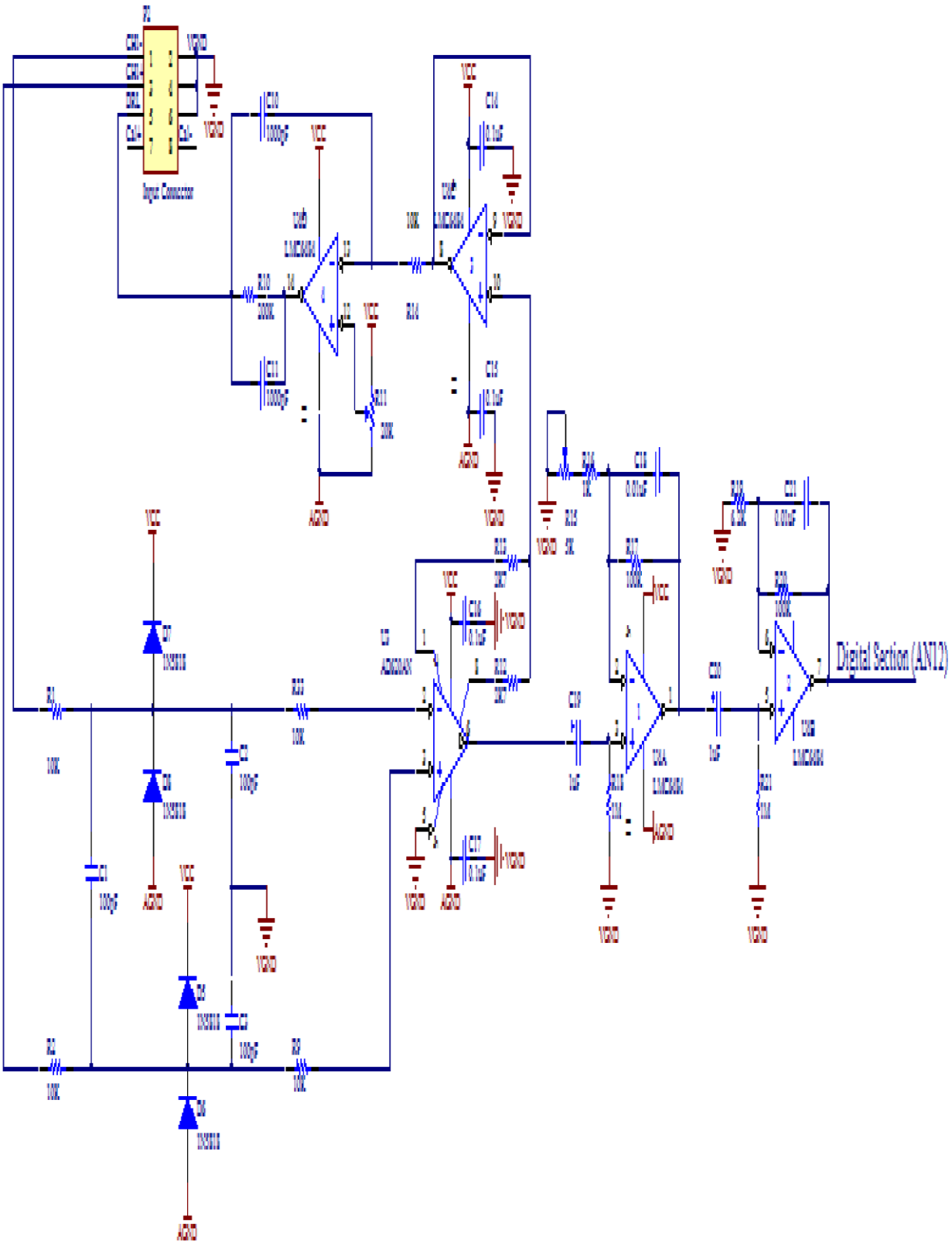
54. A.J. Rowan, E.T., *Primer of EEG.* Elsevier, Philadelphia, 2003.

55. Kubler, A., et al., *Brain-computer communication: unlocking the locked in.* Psychol Bull, 2001. **127**(3): p. 358-75.

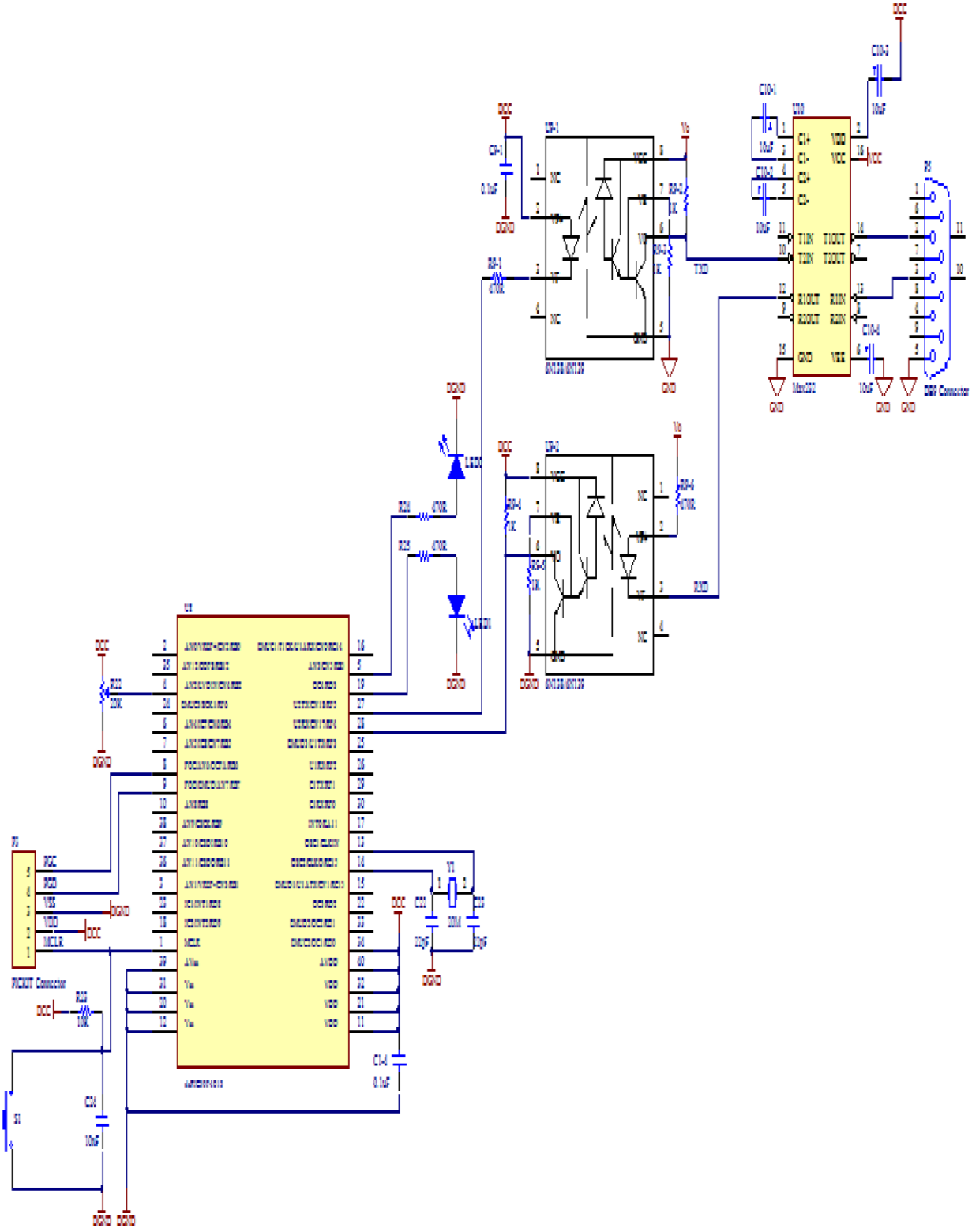
56. Carmo, J.P., et al., *A Wireless Eeg Acquisition System with Thermoelectric Scavenging Microdevice.* Biodevices 2009: Proceedings of the International Conference on Biomedical Electronics and Devices, 2009: p. 380-383.

57. Leonov, V. and R.J.M. Vullers, *Wearable electronics self-powered by using human body heat: The state of the art and the perspective.* Journal of Renewable and Sustainable Energy, 2009. **1**(6).

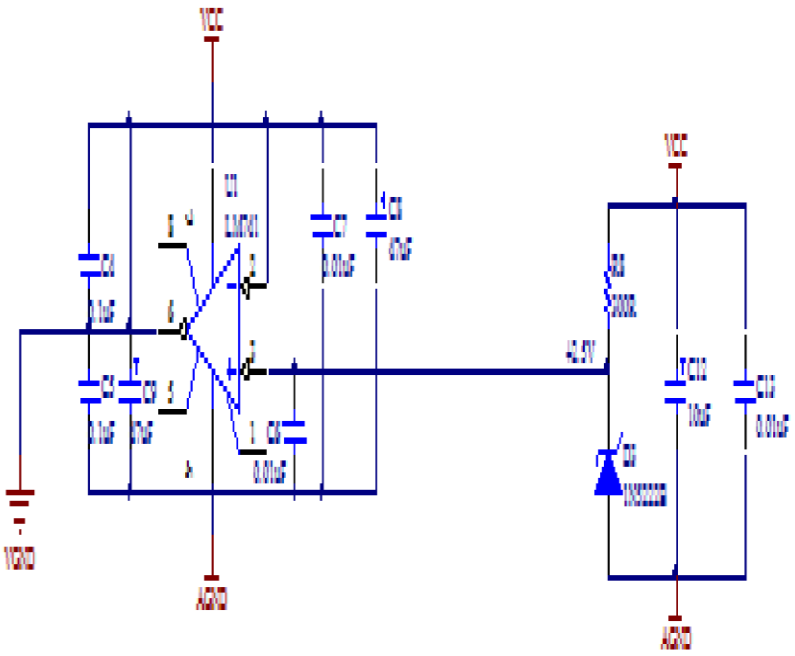
APPENDIX A: ANALOG SECTION

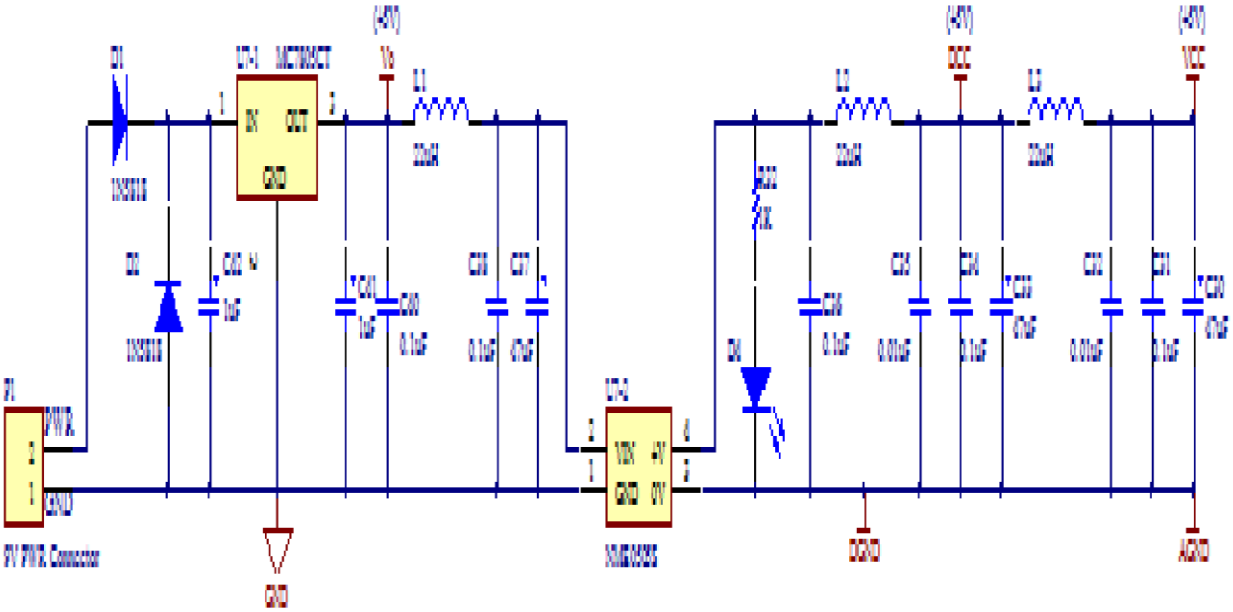


APPENDIX B: DIGITAL SECTION



APPENDIX C: POWER SUPPLY SECTION





APPENDIX D: BASICS OF A COMPUTER SYSTEM

Components of a computer

A basic computer is built up of hardware and software which are discussed as follow:

Hardware: Computer Hardware consists of the physical components of the computer. Hardware of the computer consists of I/O, memory and CPU

Input/ Output (I/O) devices (or peripheral devices) are the means by which computer communicates with other computers or users.

Memory stores information (instructions (programs) and data) temporarily or permanently as bytes (8-bit data) or words (16-bit data). RAM and FLASH are most widely used memory storage devices.

RAM (Random access memory) is called volatile memory, since as soon as the power is turned off the data it's working with is lost; therefore it is used for temporary storage of information.

ROM (Read only memory) is called volatile memory, since it holds information even in the absence of power. Hence it is used for the permanent storage of essential programs information needed for proper functioning of computer. Data of ROM can only be read.

EEPROM (Electrically erasable programmable read-only memory): is a type of non-volatile memory which can be reprogrammed (it requires erasing the memory locations before reprogramming it by programmer). Here like RAM the contents of EEPROM can be changed during an operation however like ROM contents of EEPROM are permanently.

Flash memory is another non-volatile memory that can be erased and reprogrammed up to or more than 100,000(PIC18F2525 data sheet)times (practically unlimited), devices with Flash memory are ideal for initial phase of design. Flash memory is faster than EEPROM, since data in

EEPROM needs to be erased and reprogrammed one byte at a time where as Flash can erase and reprogram bytes of data at a time.

CPU is called the brain of a computer system as it controls all the components of the computer system. CPU at least consists of registers (are used for storage of data/address during execution of instructions thus fast program execution), Arithmetic logic unit (ALU) (is used to perform logical and numerical computations) and Control Unit (fetching, decoding and monitoring the execution of instructions).

Microprocessor is CPU packaged in a single integrated circuit (a chip). Microprocessors are referred as n-bit (where n= 4, 8, 16, 32, 64) microprocessors depending on the number of bits it processes in one operation. In order for it to work as a computer system it needs to access external memory, external I/O and other peripherals. Therefore these systems are more expensive, bulkier, have higher power consumption and slower execution speeds

For example: ARM processor, Intel

Microcontroller is single integrated circuit (a chip) built with CPU, RAM, ROM, I/O ports and other peripherals (such as timers, adc, usart etc). Hence it's suitable for applications where less space, low cost and lower power consumption is needed.

For example: five major 8-bit microcontrollers are PIC from Microchip Technology, Intel's 8051, Atmel's AVR, Freescale semiconductor 68HC08/68HC11 and Ziglog Z8.

Criteria of choosing Microcontroller:

Meets the computational needs of task at hand efficiently and cost effectively.

Availability of software and hardware development tools such as emulators, debuggers, compilers, stimulators etc.

Number of bits for Microcontroller (8-bit, 16-bit, 32-bit, 64-bit), the number of bits the Microprocessor can manipulate in one operation. More the number of bits, faster is the execution speed.

Pin count(number of I/O Pin available or number of I/O ports available); if you want to interface more peripheral devices you need more I/O pins (Microcontrollers range for 16 pins to 100 pins).

Memory size, amount of RAM, ROM and EEPROM in the microcontroller

Maximum frequencies, faster the clock speed, faster is the program execution

For example, for a 20 MHz crystal Tosc = 50ns an instruction is executed every 50 nanoseconds. Maximum frequencies for microcontrollers range from few KHz to hundreds of MHz selection of frequency depends on the application.

Ease of upgrade, an application written in 8-bit PIC18F2525, works in PIC18F4560 as well as dsPIC30F families just with few changes and also the wide availability.

Software gives a set of programs (instructions) to the CPU to perform specific task/s. Program is stored in the computer memory as binary numbers called machine instructions. Also CPU works only with binary numbers, therefore all programs need to be converted into machine instructions for execution.

Thus programming, debugging and maintaining (user who didn't write the machine instructions will have difficulty in reading it) in machine is extremely difficult and prone to errors.

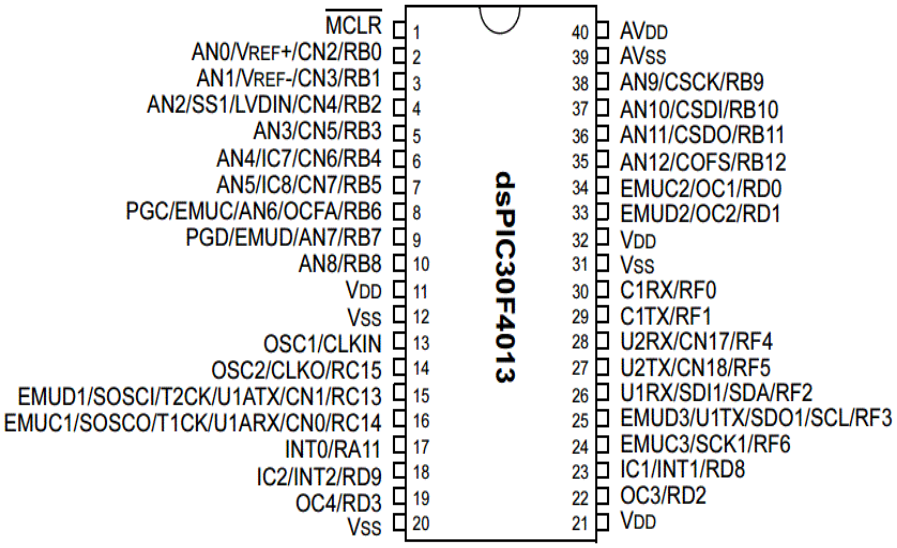
Hence Assembly was developed, which provided mnemonics (codes and abbreviations easy to remember, for machine code instructions, which made execution faster and less troublesome. Programs here are converted into machine code by assembler. Assembly language is also called low-level language and here the programmer needs to have a thorough knowledge of the internal structure of CPU. Today users can program MCU's through high-level Languages such as

BASIC, C, C++ and JAVA without worrying about internal architecture of the CPU, thus programs here are easy to write and debug making programming easier. Programs in high-level languages are converted into machine instructions using compilers. One of the drawbacks of using high-level languages coding in requires much more machine instructions than assembly languages. Hence programs which time-critical are still written in assembly.

APPENDIX E: dsPIC30F4013 MICROCONTROLLERS: (DATA SHEET)

We have used dsPIC30F4013 (F stands for Flash ROM memory) for our project because of the wide variety, availability and easy upgrade of Microchip PIC microcontroller (you can select the PIC microcontroller according to you specification with Microcontroller Product Selector (MPS) tool), excellent free development tools provided by Microchip such as MPSIM, MPLAB, compiler (MC30) etc, efficient execution speed (up to 30MIPS) and peripheral features such as a 12-bit A/D converter, SPI, I2C and UART. You can either program PIC microcontrollers using low-level (assembly) or a high level language (C). We have used c language for coding and C30 compiler.

The pin diagram for the dsPIC30F4013 controller is shown below:



I/O: Input devices such as keyboards, mouse etc. provides digital information to the microprocessor and the output devices such as LED, LCD etc. receive digital information from the microprocessor. These devices are interfaced to the microprocessor using the I/O ports.

DsPIC30F4013 includes five bidirectional I/O ports; they are PORTA, PORTB, PORTC, PORTD and PORTF. Pins of each PORT can be either configured individually or together as

digital inputs or digital outputs. All pins are multiplexed meaning they can be setup by writing instructions to perform one or more functions of other peripheral devices. When configured to operate with the peripheral such as A analog to digital converter, a pin cannot be used as a general input or output pin. Also at reset all the port pins are programmed as inputs.

Each port has three 8-bit SFRs associated with it they are designated as

TRISx register (data direction register, where x stands for Ports A, B, C, D and F), sets up a port as input or output. Setting a bit to 1 configures a pin as Input and Clearing a bit to 0 configure the corresponding pin as an output (Notice that 0 stands for out and 1 for in. this is easy to remember because O and 0 look alike the same way that I looks like 1).

For example:

To set up an entire port as input or output

TRISx =0b0000000,

Example: set PORTA as output port we write to TRISA register as follows:

TRISA =0b00000000;

To setup an pin of a port as output or input;

TRISxbits.TRIScy=1; where x = PORT A, B or C and y = Pin number (0-7)

Example: Set Port A pin 6 as output

TRISAbits.TRISA6 =1.

PORTx register (reads the levels on the pins of the device), is used to read the input of the actual voltage level applied to the pin from the configured PORTs. PORT register requires the PORT to be configured as input by using the TRIS register.

For example: Suppose we have a switch on pin 0 of PORTA which is configured as input, than if

We read from PORTA when the switch is pressed we get

PORTA= 0b00000000;

PORTAbits.RA0 =0;

Or we get the following when we read from PORTA when the switch is released.

PORTA= 0b00000001;

PORTAbits.RA0 =1;

LAT register (output latch), it is the output latch onto which values are written. And this register is independent of the configuration of the TRIS register.

For example: If a result of a particular program when correct switches ON an LED on pin 3 of PORTC

Than LATC= 0b00000100; //Pin 3 is high

Or LATCbits.LATC3 =1;

Mirochip support staff recommends to use PORTx for reading data in and LATx for writing data out - especially when using single bits for I/O.

APPENDIX F: UART PACKETS INTERFACING WITH HYPERTERMINAL, MATLAB AND BCI2VR

Testing of the dsPIC30F4013 transmission module can be done in the following ways:

Communication with the PC using HyperTerminal

1. To launch HyperTerminal

Start → Program → Accessories → Communication

2. When the HyperTerminal's is launched Connection Description window appears, to set up the program enter in name for this communications session.

3. A HyperTerminal's Connect to screen window appears, choose the serial (com) port you are using from the Connect Using field. Click OK.

4. Next HyperTerminal's COM1 (port you have selected) Properties window appears. Select values for the fields according to your configuration. For example for our UART packets configuration, the values in the following fields would be:

Bits per second: 115200 (baud rate of the computer), Data bits: 8, Parity bits: none, Stop bits: 1

Flow control: none

5. Check the ASCII setup from HyperTerminal:

- From the File menu, select Properties.
- In the Properties screen, select the Settings tab
- In the Emulation field, select "ANSI."
- Click on the ASCII Setup button.
- In the ASCII Setup screen uncheck all the boxes.
- Click OK in the ASCII Setup screen.

- Click OK again the Properties screen.

6.. If the PIC MCU transmit program is running, the string sent to the PC by the console program should be displayed in the HyperTerminal window.

Communication with the PC using Matlab

In the matlab command prompt type the following:

1. s = serial('com7'); constructs a serial port object associated with port, PORT you are using . If PORT does not exist or is in use you will not be able to connect the serial port object to the device. For example use cannot Interface USART packets with Matlab and HyperTerminal at the same time.

2. set(s, 'BaudRate', 115200); set the baud rate of the interface (replace it with the baud rate you are using).

3. fopen(s); connects the serial port object to the serial port.:

4. get(s); returns the information of the packets (such as packet order, number of bytes available).

5. fread(s); reads and displays the data.

6. fclose(s); disconnect the serial port object from the serial port in other words closes the connection.

Communication with the PC using BCI2VRMatlab Toolbox

1. In the BCI2VR GUI interface, select Data Acquisition
2. File-> load set-up file -> (C: SharedFolder->BCI2VR_Toolbox->setup_files->rs232_EBLAmp_btv_daq_settings.m) -> Open.
3. Run calibration (the green button) and record.

VITA

Dimple Chitranjan Bhuta was born on January 5th, 1988 in Mumbai, India and is a citizen of India. She received her Bachelor of Engineering degree in electronics engineering, from K.J. Somaiya College of Engineering in Mumbai, India in 2009. In August 2009, she joined the School of Engineering at Virginia Commonwealth University to pursue her Master of Science degree in the field of Biomedical Engineering. She was invited to be a member of Phi Kappa Phi Honor Society and was a mentor in the Graduate School Mentorship program (fall 2010 -spring 2011).At the Bioinstrumentation and medical imaging lab, she worked on her thesis project titled 'Brain Controlled Switch' under the guidance of Dr Ding-Yu Fei during the course of her degree. Currently she is pursuing her Master of Science degree of which the above thesis is a partial requirement.