SYSTEM REQUIREMENTS SPECIFICATION

CS101 EMBEDDED PROJECT 2015

MORSE CODE DECODING & PROCESSING

"Morse Code Bot"

Group 365

1.Ajinkya Gorad	140110033
2.Kewal S. Bhat	14D110008
3.Shrenik S. Jain	140110045
4.Jenil P. Shah	14D170008

CS101 SPRING 2015

INSTRUCTOR : KAVI ARYA

This page is intentionally left blank

Abstract

This document describes the technical aspects of the project developed on Firebird V based on ATmega2560.The project titles ‘Morse Code Decoding & Processing’ and the robot named Morse Code Bot. It aims on receiving the morse code sequence and act accordingly.

Contents

Abbreviations

uC	**Microcontroller**
ACK	**Acknowledge**
WPM	**Words Per Minute**
ADC	**Analog to Digital Converter**
CPU	**Central Processing Unit**
CW	**Continuous Wave**
IR	**Infra Red**
MIC	**Microphone**
DSP	**Digital Signal Processing**

1. Introduction

Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks. Transmitting information that only skilled persons understand it. Controlling any unit such as robot with a coded language is noticeable even when transmission is done through audio using coded language called Morse Code.

This project aims Morse Code Controlled Bot controlled through audio , i.e. a series of on-off tones. CW morse code transmission is used.

1.1 Background

Morse code consist of sequences of pulses called dits and dahs determined by their timeperiod. A dah's duration is three times a dit duration. Morse code is combination of dits and dahs separated by duration of dit, i.e. inter pulse spacing is duration of dit. Also inter letter spacing is of duration of 3 dits or a dah and inter word spacing is of 7 dits. WPM of Morse code is given as : 1200/T, where T is the dit time period in milliseconds.

The bot will receive the morse code and take action as defined in bot's software. For this project bot hears the morse code and takes actions move forward, backward, right, left or to move on a polygon of sides 3 to 9.

If the morse code transmitted is `..-.` (dit dit dah dit) or 'F'

The bot will move forward.

A single digit number n is used to describe the bot to move in n sided polygon

If the morse code transmitted is ...-- (dit dit dit dah dah) or '3'

The bot will move on triangle.

After executing the command bot will acknowledge by sending "ROGER" in morse code.

`

1.2 Structure

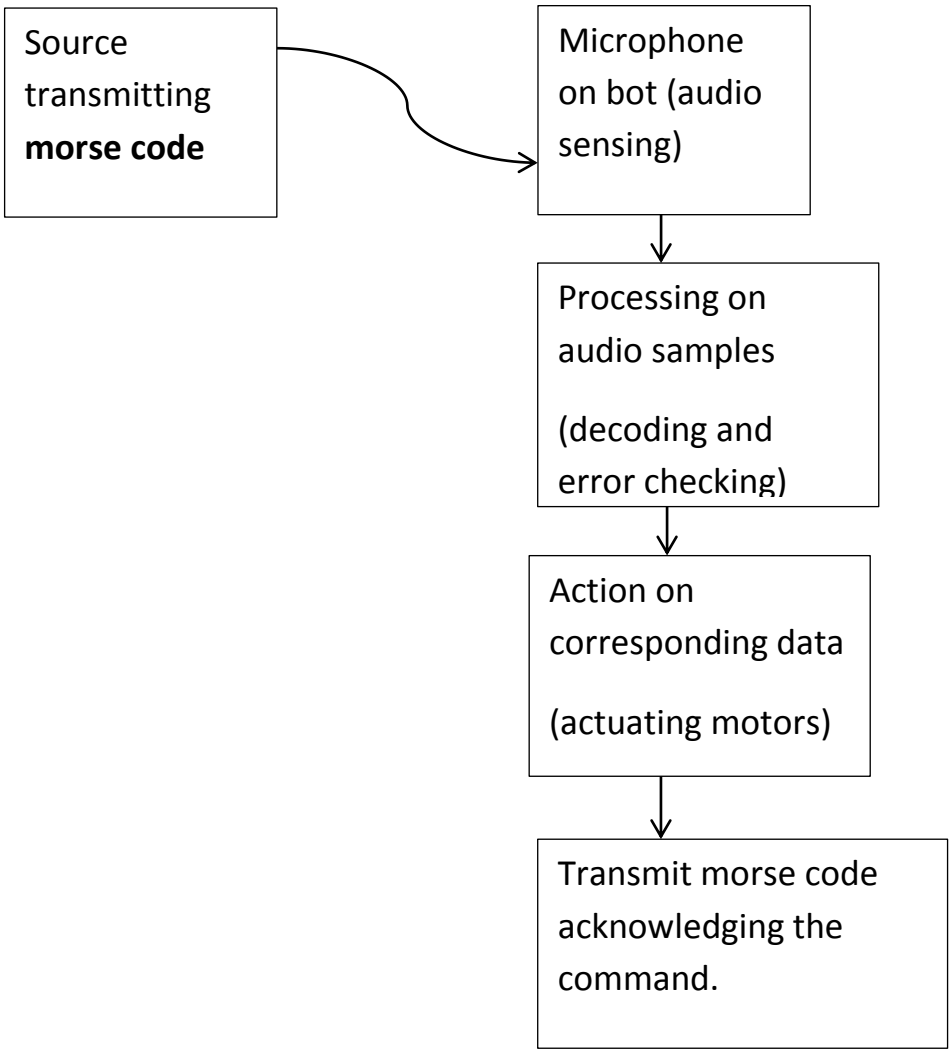

Fig 1.0. Basic structure of design implementation

2. Problem Statement

Making of the bot's additional hardware and software divided in 4 parts:

1. Using condenser mic to read audio signals and providing it to the microcontroller. Decoding audio information into corresponding units of Morse Code as dits and dah.

2. Using the dits and dahs to find the corresponding letter.

Eg. -... correspond to 'B'

3. Processing the corresponding letter to perform mobile operations by controlling motors.

4. Acknowledging the command sent by user by transmitting the morse sequence of "ROGER " using on board buzzer.

3. Systems and Requirements

3.1 Hardware

The external (non software) requirements for Morse Code Bot are as follows:

1. Firebird V –Atmega2560 (as the bot)

2. Condenser mic module (ref. Fig 4.1)

3. Morse Code Transmitter & Receiver (eg. Smartphone).

However , the similar can be implemented on independent components using

1. uC having at least two timer peripherals.

2. Condenser mic. (with circuitry to amplify the signals) (ref. Fig 4.1)

3. Actuator motor with their drivers.

4. Buzzer (to transmit morse code by bot, ACK-ing the sequence)

5. Battery (as power source)

3.2 Software

Software Requirements on PC are as follows:

1. Atmel studio 6.0

2. AVR bootloader serial programmer (for firebird V)/ or any programmer for respective device . eq. ISP with usbasp for AVR.

3. Morse Player (http:// morsecode.scphillips.com/translator.html) as transmitter.

Software part is written in C++ using object oriented programming.

Software Requirements on Android Smartphone (if used)

Android app from Play Store :

1. Morse Code Reader (receiving morse code from audio).

2.Morse CT (Morse Code Trainer).

4. Implementation

4.1 External Hardware and Module 1

Following circuit was used for making of the condenser microphone amplifier and getting the kind of input to easy to process with it.

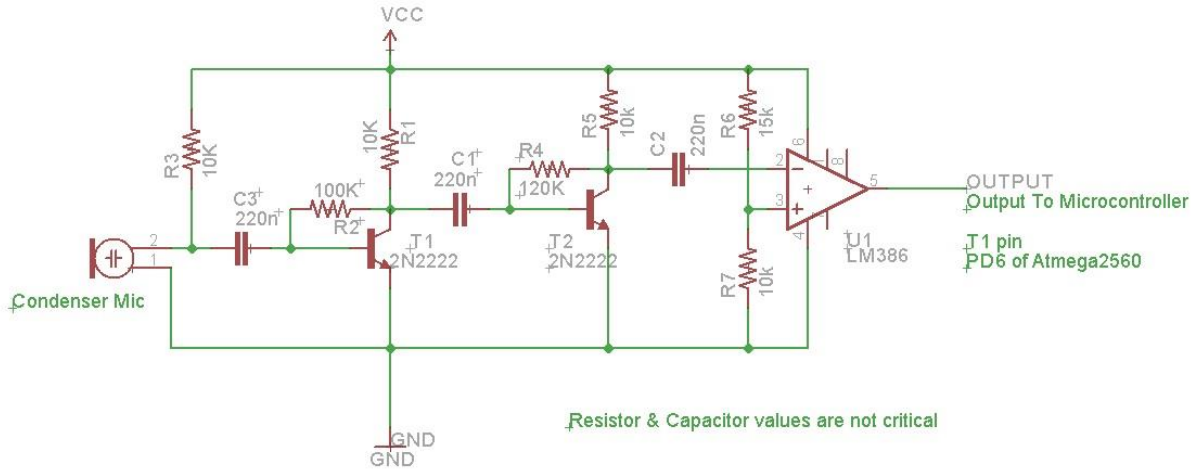

Fig 4.1 Circuit Diagram for microphone

This circuit first amplifies the audio signals. The output of amplifier is fed to the comparator on the right which provides the square wave pulsed signal which is given to the microcontroller.

Processing through microcontroller :

Following flowchart (fig) describe the software part of the receiving sequence of audio data and finding the dit or dah and storing it.

uC sees a square wave signal on T1 pin (on firebird V) of ATmega2560, counts the number of falling edges until no next falling edge arrives at specific time and is defined as timeout for edge. Fig 4.3 detail about capturing of data sequence for a given pulse & Fig 4.4 finding dit & dah from the data captured by capturing sequence. Data seen by the microcontroller is like Fig 4.2

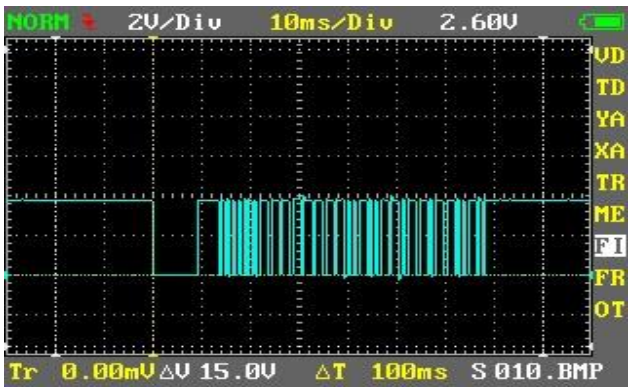

Fig 4.2 Output from Comparator (similar to)

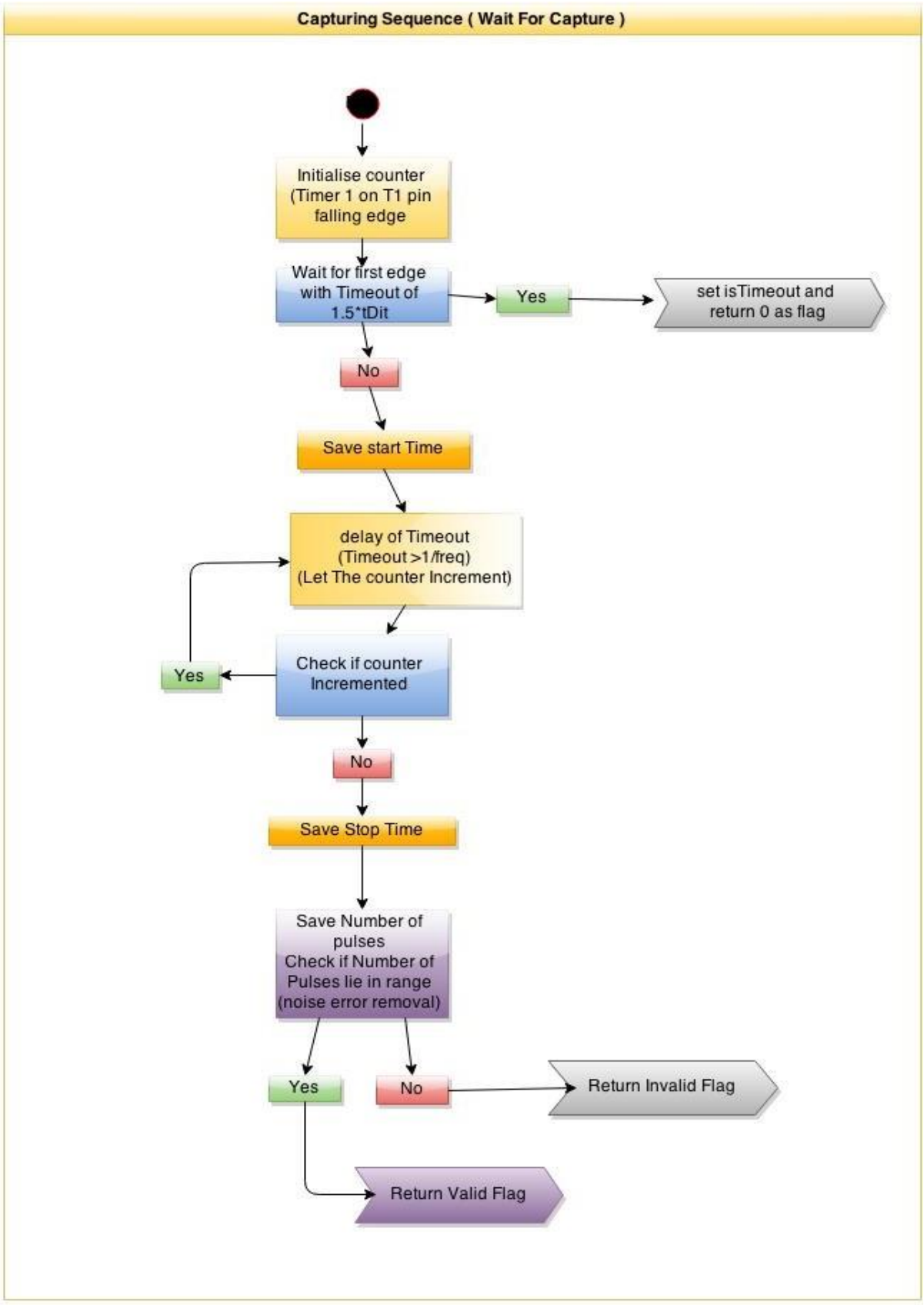

Fig 4.3 Flowchart for capturing data

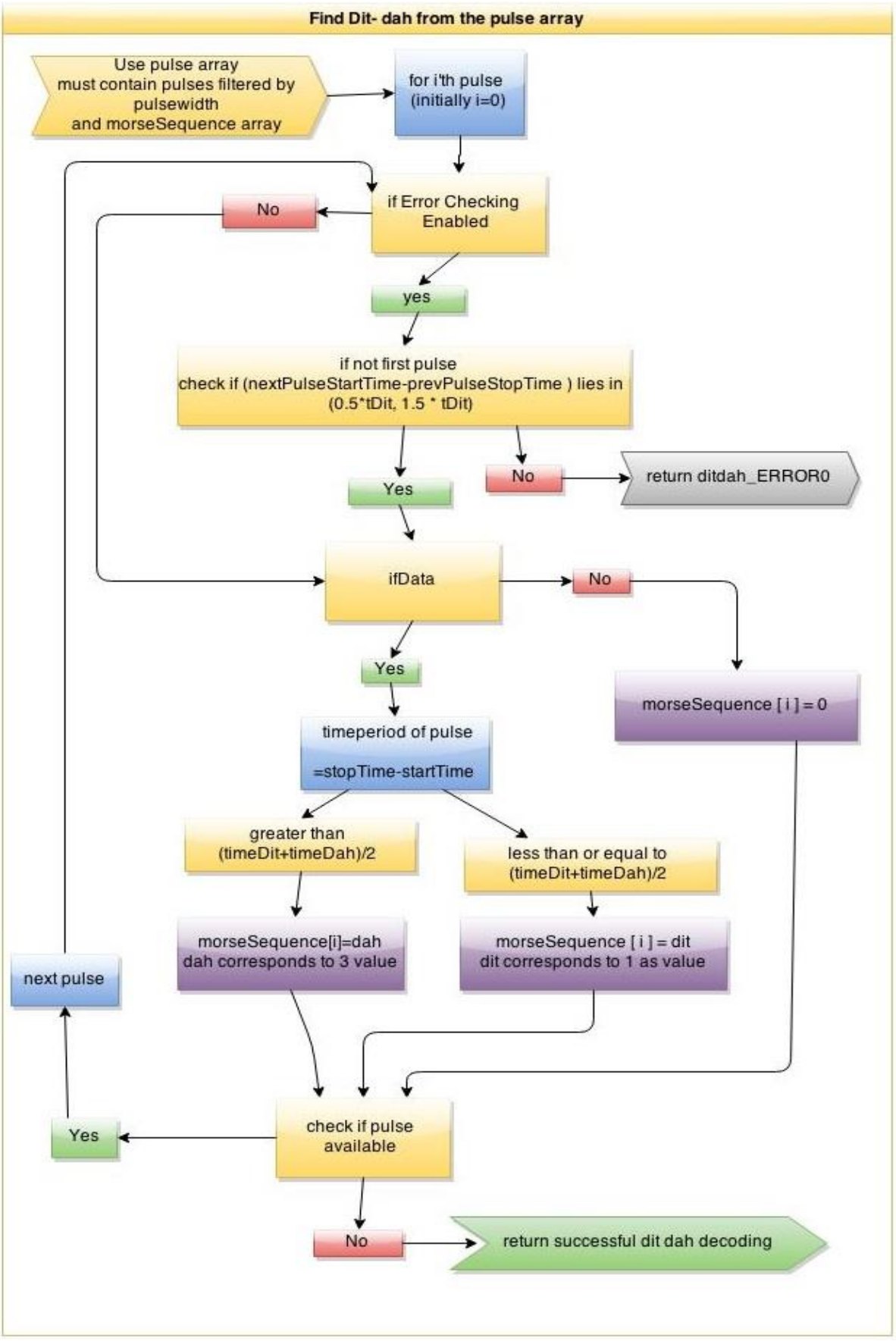

Fig 4.4 Flowchart for finding dit or dah from pulse data

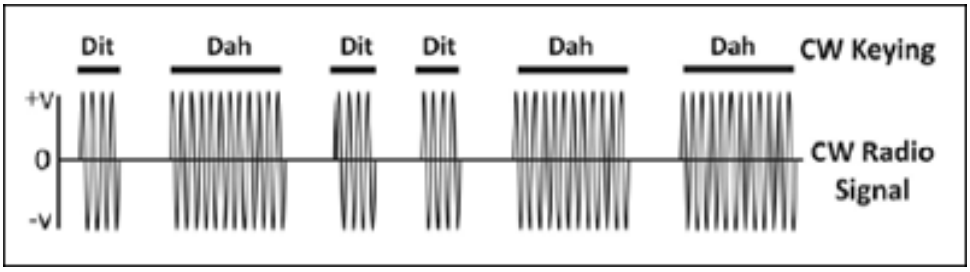

Fig 4.5 CW keying morse code

4.2 Module 2

Finding of corresponding letter from dit dah array.

Binary tree of morse code is used to find character from dit dah sequence.

Algorithm works by assigning the numbers in order as shown and storing it in an array .

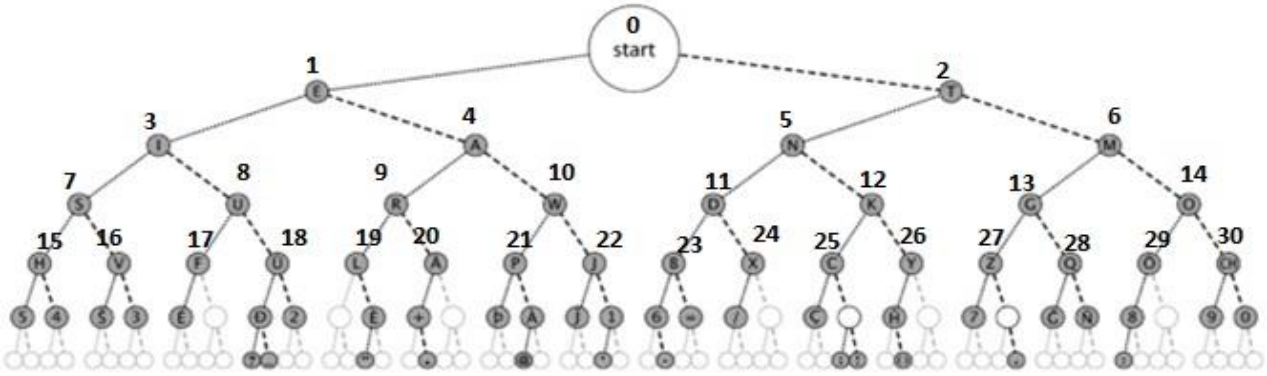

Fig 4.6 Morse Code Tree (edited, source Wikipedia)

Initially data pointer is at 0. If the next data found corresponds to dit then pointer is moved to (2*n+1) th position or if next data found corresponds to dah then it is moved to (2*n+2) th position, where n is the current index (initially 0). Dit is stored as ASCII ‘1’ and dah is stored as ASCII ‘3’.

When no data for dit or dah is found i.e. it has completed the branching it will return the data corresponding to current position.

For eg. If data was `..--` (dit dit dit dah dah) it will correspond to index of 18 with letter ‘U’

4.3 Module 3

From the letters decoded the bot will have to decide what to do with it.

In this project only simple functions such as moving forward, backward, right, left or moving in a polygon are implemented. However complex function with additional hardware or programming could be implemented.

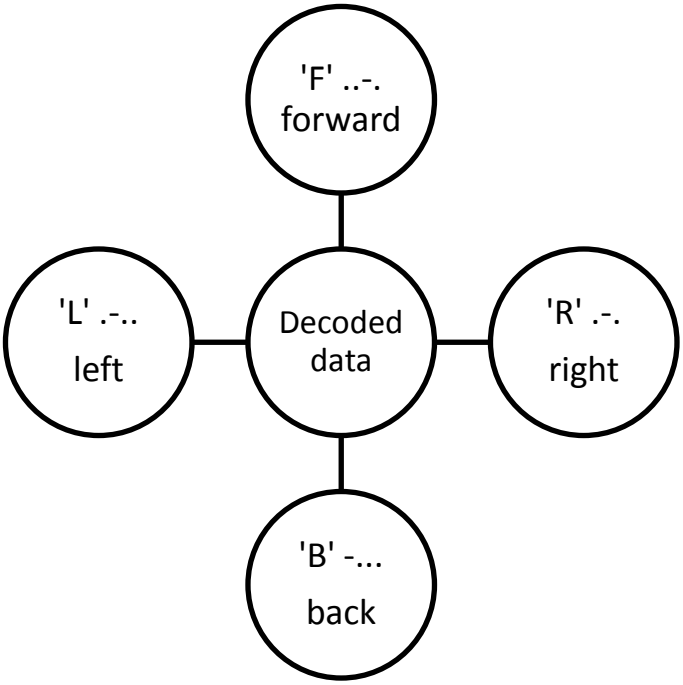

Fig 4.7 Movement command for bot

For moving on a polygon decoded data must be in the range of ‘3’ to ‘9’.

Motors used with encoders to have the feedback of how much the motor shaft have rotated. Algorithms/code have been adopted from eYantra for firebird V.

4.4 Module 4

After performing the required operation to notify the user that respective operation has completed, the bot will acknowledge the user by transmitting “ROGER” to signify the end of operation.

ROGER in morse code

.-. --- --. . .-.

R O G E R

This is done using on board buzzer on Firebird V.

5. User Characteristics

To control the bot user must have the morse code transmitter to give commands to the bot at required WPM. Also a receiver if user is not a skilled listener.

Nominal rate of morse code transmission used is 30 WPM but can be adjusted in software up to 70WPM.

Nominal frequency of morse code transmission is 1000Hz but it is fine if frequency range varies between 700Hz to 2000Hz.

User can use the following site to transmit or receive morse code from PC or any device which supports internet :

1. To transmit : http://morsecode.scphillips.com/translator.html
2. To receive : http://morsecode.scphillips.com/transcriber/

http://morsecode.scphillips.com contains more tools regarding morse code.

6. Limitations

Can only receive CW modulated morse code signals

Bot fails to receive morse code instructions in noisy environments due to the sensing hardware and crude processing of data.

Range of reception is very low ~ 1.5 feet in low noise environments due to crude design of mic amplifier.

7. Future Work

1. Algorithm can be modified for more CW waveform protocols to receive and transmit information in coded form like in IR Remote Control.

2. Instead of mic, one can use radio waves to transmit morse code and control bot over large distances.

3. Accuracy can be improved by using ADC to sample the data and process it using DSP algorithms at the cost of CPU processing and power.

4. A tone decoder like NE567 can be used to hear morse code at particular frequency adding frequency dependent noise filtering.

8. Online Support

Code can be found on github repository
https://github.com/ajinkyagorad/CS101_MorseCodeDecoder

9. Sources

1. http://morsecode.scphillips.com

2. http://en.wikipedia.org/wiki/Morse_code

3. For drawing flowcharts https://www.draw.io/

4. For drawing schematics using Eagle http://www.cadsoftusa.com/

5. Motivation
http://www.hamradio.cc/projects/Morse_Code_Decoder_Circuit.php

10.References

1. For getting started with Firebird V

http://www.e-yantra.org/index.php/eyantra/tutorial

2. Firebird V code sources from eYantra

https://www.dropbox.com/sh/2orydpmoopeczi3/AAARr6-jAeD7zlTCYq4rChZCa

3.For microphone amplifier circuit

http://www.scienceprog.com/electret-condenser-microphone-amplifier-for-use-in-microcontroller-projects/

4. Using morse code to type (iambic keyer) , i.e. as a keypad

http://iditdahtext.com/iDitDahText.html

5. Github repository

https://github.com/ajinkyagorad/CS101_MorseCodeDecoder

6. Learning more about Morse Code and transmitter receiver applications

http://morsecode.scphillips.com