Limitations

1. Due to unavailability of huge training set and high speed training machines, the accuracy is not as expected i.e around (40-60)%
2. Only five words can be trained to the system
3. The neural network contains only 1 hidden layer with 50 nodes and hence it cannot be used for the large scale modelling
4. Response time is quite large i.e 2-3 seconds per instruction.
5. Noise in audio signals could not be filtered efficiently (Microphone noise and environmental noises)
6. The Bluetooth connection is not stable enough
7. The movement of the motors in not symmetrical

Output:

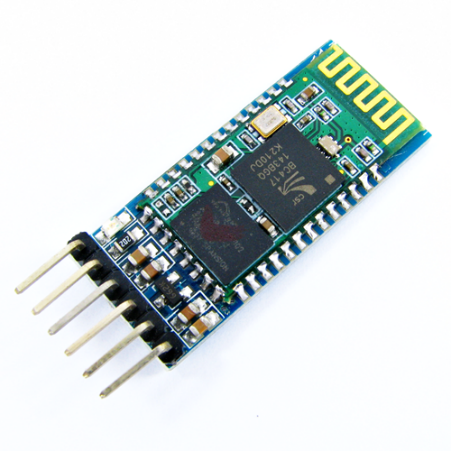
The output layer is a softmax layer and there are five output nodes. The softmax layer is the normalized exponential function and is used fo the probabilistic multi-class classification in neural network.

In our system the mean softmax output of the recording is correlated with the ideal output to classify the instruction.

Communication of PC and hardware:

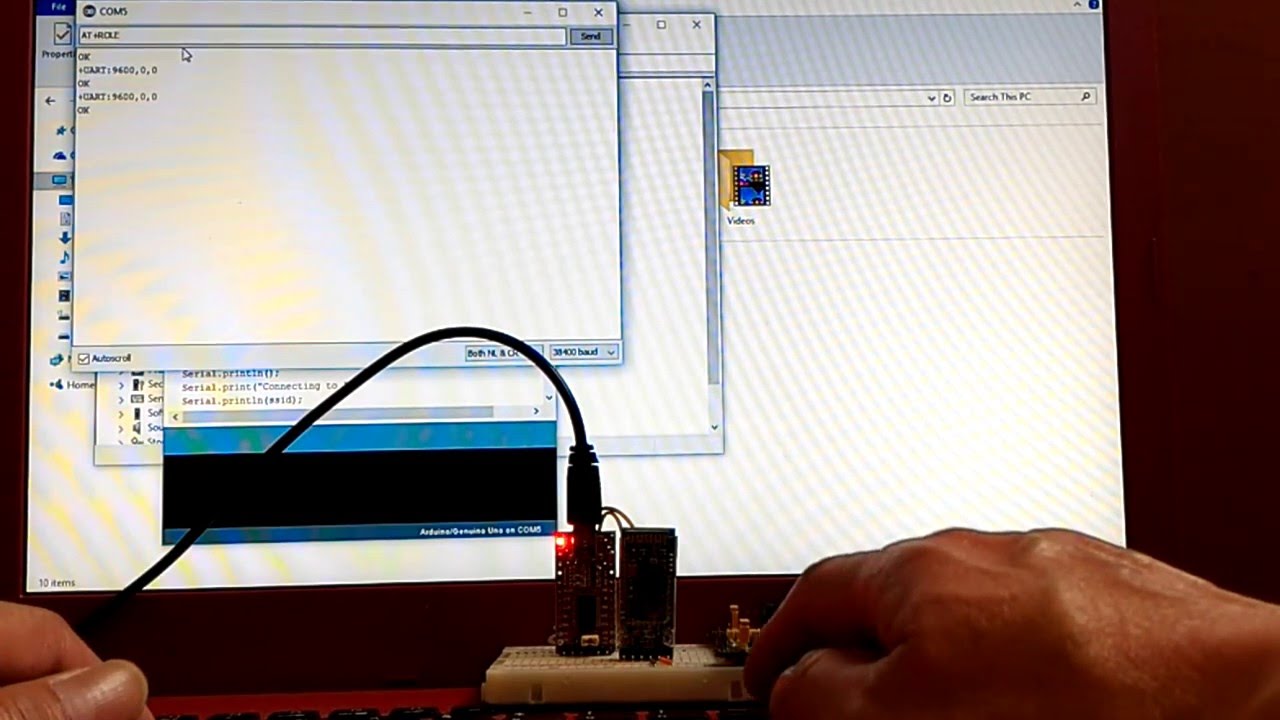
After the speech is recognized, the corresponding instruction is passed from the computer (PC) to the robot car. The communication is done through the Bluetooth of the PC

On the hardware end the data is received via HC-05 Bluetooth module



The baud rate was set 9600 with the parity bit set to none and one stop bit.

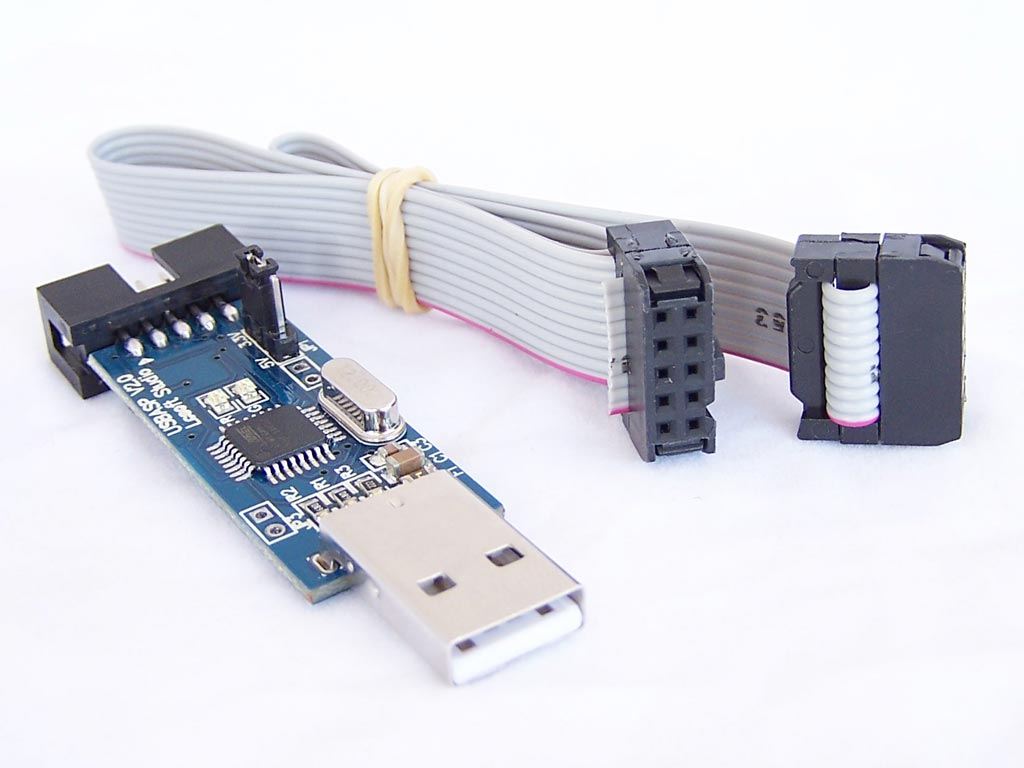
The HC-05 was set up via AT-command of the module where the name and the security settings of the module was configured.



Hardware:

ATMEGA32 was used as the microcontroller for the robot car. We clocked the ATMEGA for the operation in 16MHz using the external crystal.

The programming was done in C programming languages using the atmel studio and was burned in ;the ATMEGA using the USBASP programming module



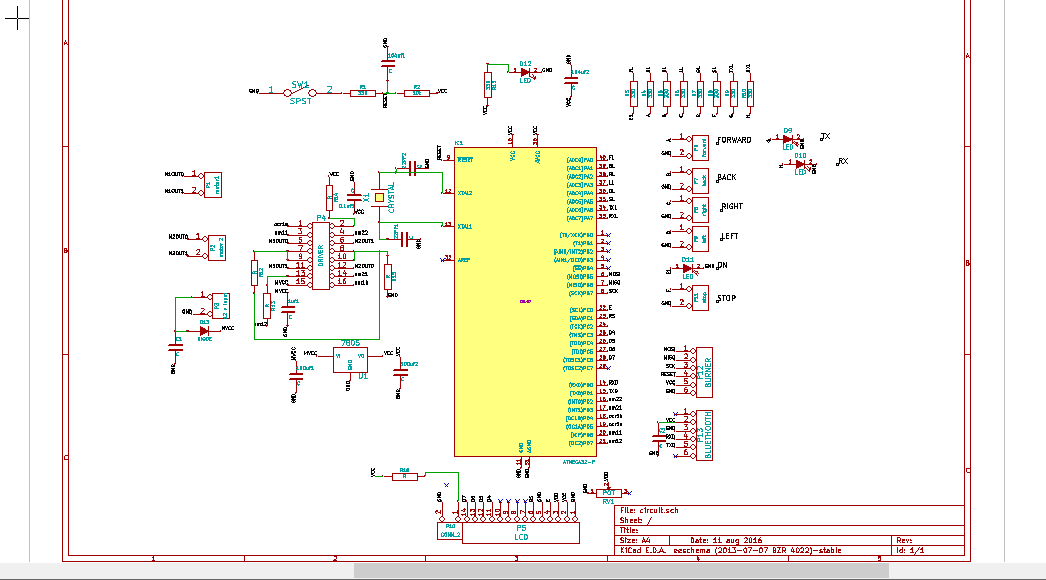
The programming includes the reception of the serial data via Bluetooth module and pass the pass the PWM values accordingly to the L293D motor driver IC.

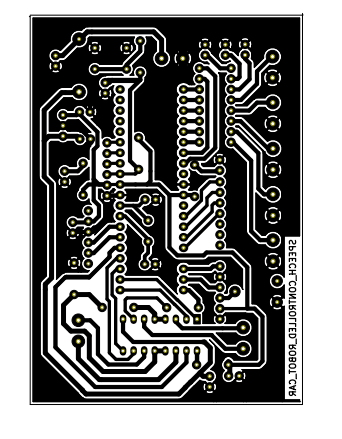
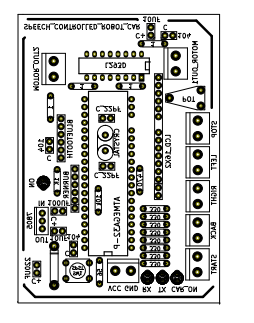
Then the motor driver IC drives the motors (wheels) according the data provided by the microcontroller



PCB and shouldering

The pcb was designed using the KiCAD first in schematic and later in the PCB. The track width was set 0.7 mm and the remaining copper are was selected as the ground plane





Introduction USART

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:  
• Full Duplex Operation (Independent Serial Receive and Transmit Registers)  
• Asynchronous or Synchronous Operation  
• Master or Slave Clocked Synchronous Operation  
• High Resolution Baud Rate Generator  
• Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits  
• Odd or Even Parity Generation and Parity Check Supported by Hardware  
• Data OverRun Detection  
• Framing Error Detection  
• Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter  
• Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete  
• Multi-processor Communication Mode  
• Double Speed Asynchronous Communication Mode

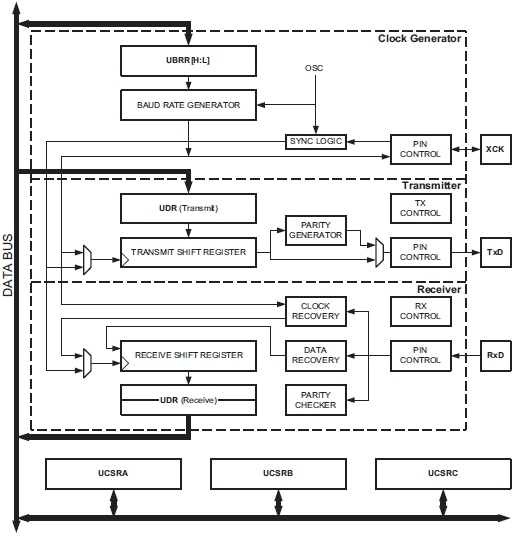


Fig: Block Diagram of USART

The dashed boxes in the block diagram separate the three main parts of the USART : Clock Generator, Transmitter and Receiver. Control Registers are shared by all units.The clock generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by Synchronous Transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery-units. The recovery units are used for asynchronous data reception. In addition to the recovery-units, the receiver includes a parity checker, control logic, a Shift Register and a two level receive buffer (UDR). The receiver supports the same frame formats as the transmitter, and can detect frame error, data overrun and parity errors.

**Internal Clock** **Generation – The Baud Rate Generator**

Internal clock generation is used for the asynchronous and the synchronous master modes of operation. The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (fosc), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= fosc/(UBRR+1)). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the receiver’s clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSEL, U2X and DDR\_XCK bits.

For asynchronous normal mode,

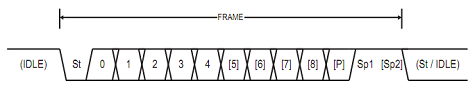
BAUD = fosc**/**16(UBRR+1)

For synchronous master mode,

BAUD = fosc/2(UBRR+1)

Where fosc refers system oscillator clock frequency

**Frame Format**



The frame format used by the USART is set by the UCSZ2:0, UPM1:0, and USBS bits in UCSRB and UCSRC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter. The USART Character SiZe (UCSZ2:0) bits select the number of data bits in the frame. The USART Parity mode (UPM1:0) bits enable and set the type of parity bit. The selection between one or two stop bits is done by the USART Stop Bit Select (USBS) bit. The receiver ignores the second stop bit. An FE (Frame Error) will therefore only be detected in the cases where the first stop bit is zero

PWM

Timer/Counter0 is a general purpose, single compare unit, 8-bit Timer/Counter module. The  
main features are:

• Single Compare Unit Counter

• Clear Timer on Compare Match (Auto Reload)

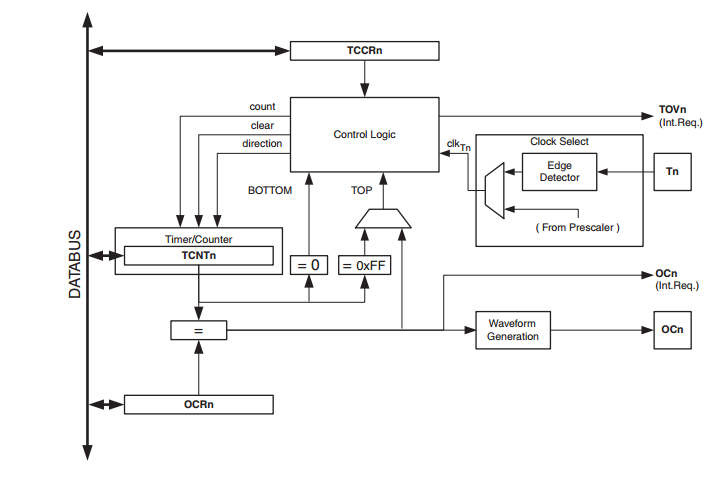
• Glitch-free, Phase Correct Pulse Width Modulator (PWM)

• Frequency Generator

• External Event Counter

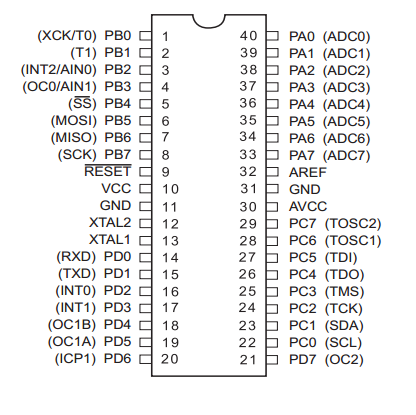
• 10-bit Clock Prescaler

• Overflow and Compare Match Interrupt Sources (TOV0 and OCF0)



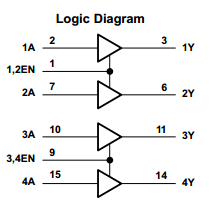
Microcontroller:

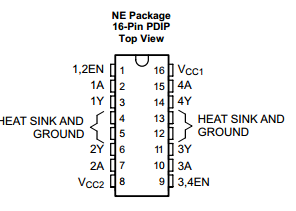
The Atmel®AVR®ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR  
enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the  
ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to  
optimize power consumption versus processing speed.



L293D

L293D is the quadruple high-current half-H driver. It provides the bidirectional drive current up to 600mA from 4.5 to 36 V. They both drive the inductive loads like relays, solenoids and motors( DC, stepper)





Feature extraction:

We extracted the MFCC of the time domain framed audio signal. Basically the MFCC extraction involved the following steps for the each frame:

1. Calculate the periodogram of the power spectrum of the frame.
2. Apply the Mel-filter in the periodogram and sum the total energy in each filter.
3. Take the logarithm of the filter banj energy and find the DCT of the value.
4. Take the first 13 coefficients and then find their corresponding delta and delta-delta.

Mel-scale

The formula for converting from frequency to Mel scale is:

http://practicalcryptography.com/media/latex/369d64804e572729863c874aaa092e582bf5eb56-11pt.png

To go from Mels back to frequency:

http://practicalcryptography.com/media/latex/05d74bc31f4c2a9c375dd9c95d4642d558f455a0-11pt.png

1. First of all we gnetate the FFt of the signal using the following formula:

http://practicalcryptography.com/media/latex/c970f070a776e4c900bd3e8a2082a2971236b013-11pt.png

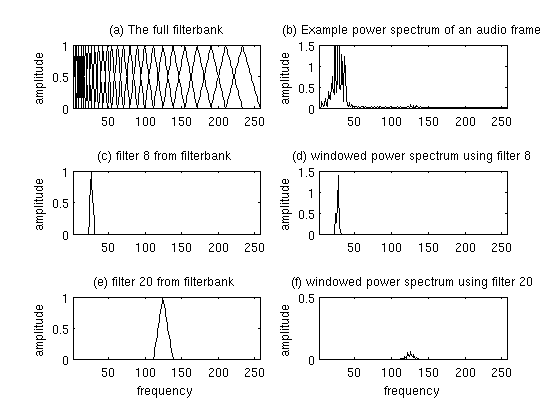
We generally obtain 512 point fft and only consider 257 coefficients.

Now we obtain the periodogram of the signal as:

http://practicalcryptography.com/media/latex/c526edb9d52e631812798237ea3f2beea496d181-11pt.png

Where N is the number of samples ( In our case the value of N is 400 I,e the size of hamming window)

1. We compute the Mel scale triangular filters as follows by converting the frequency into 26 equally spaced Mel Scale(see above formula).



The periodogram of the power is then passed through the filter and then added

1. The Logarithm of the 26 energies is then taken.
2. The DCT is applied to the log-energy value and lower 13 MFCC values are selected.
3. The deltas and delta-deltas is then calculated using the following formula:

http://practicalcryptography.com/media/latex/542b8743573ec3ff3ddbfd965512d484bc1a1818-11pt.png

The MFCC, delta and delta-delta values are then cascaded to give the feature vector of 39 elements for one frame.

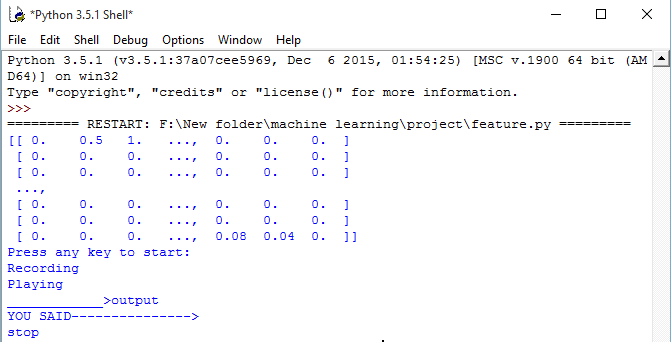


Fig ; output

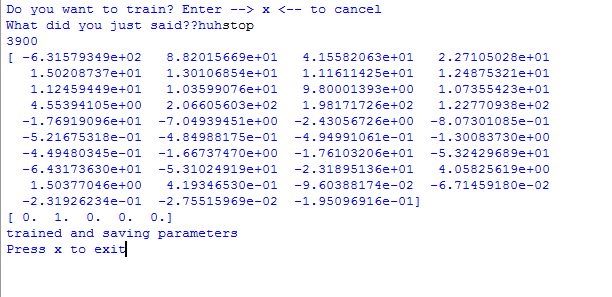


Fig: Training