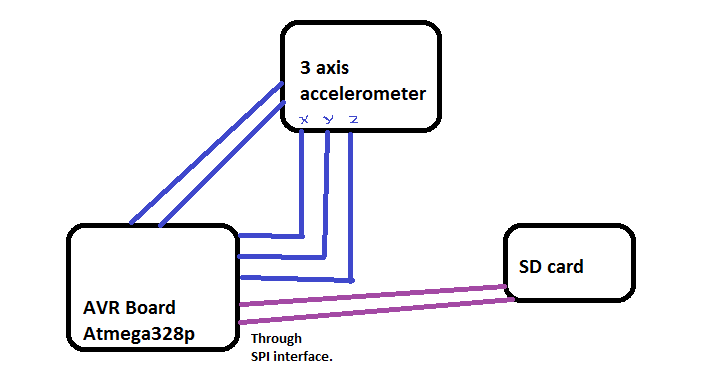
**INDIVIDUAL PROJECT REPORT: Sumit Arora(2008052)**

**Synopsis  
Aim**: To interface an external accelerometer to the microcontroller and use the device to find the acceleration of a car. The acceleration values are then plotted to analyze the driving pattern of the car. This can be further extended to find the average velocity and the fuel consumption of the car.

In the project I have interfaced a 3 axis accelerometer MM7260Q with Atmega 328P microcontroller to get the accelerometer values at rate of 2 readings per second. The values at each second are stored in a micro SD card connected using SPI interface. The device can be deployed in car and can store the values of acceleration in a journey. The user just takes out the micro SD card and plugs it on the laptop. The data is fed into an application that creates a graph and gives the analysis of the driving of the driver. The aim of the project is to give feedback to the driver about his driving pattern and find the fuel consumption for the journey. This would help the user save fuel as he would take care of his driving style as he is now aware of the fuel consumption given a particular driving style. If someday he gets really high fuel consumption he would surely learn to save fuel the next time. Just giving the feedback to the driver is good enough to help him improve driving skills and save fuel.   
The project can further be extended to give in depth details of the driving pattern of the driver. It can also be used to give warnings to the driver if he is using the phone while driving.  
I would love to make more models for analyzing the in depth driving style of the driver and then deploy it in my own car. This would really help me improve the mileage of my car.   
  
**Architecture**

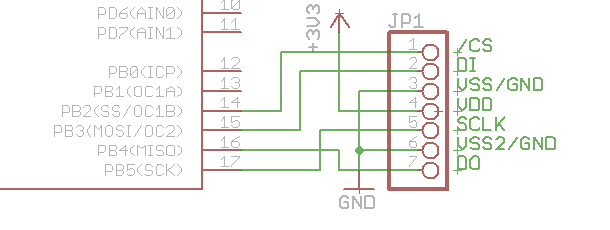
The architecture diagram of the device is shown in the following figure. It consists of the Accelerometer which is connected using the ADC pins. The Micro SD card is connected using the SPI interface.

 **Figure1: Architecture Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **HARDWARE** | **COST ( Rs.)** | **PROCURED FROM** | **Connecting with Microcontroller** |
| Accelerometer MM7260Q | 1250 | www.Rhydolabz.com | Using ADC Pins |
| Micro SD Card Breakout Board with Voltage Regulator | 325 | - | SPI Interface |
| Micro SD card | 150 | Chandni Chowk | On the SD Card Holder |
| Connectors | - | Lab | - |

****

**Figure 2 : It describes the connection of Accelerometer with the Microcontroller at the IC level.**



**Figure3 : A snapshot of the IC connection of the Micro SD card with the microcontroller**

Figure 2 , 3 are taken from the datasheet for the accelerometer and the micro SD card respectively.

**Software**The software for the application is written in C. The clock frequency is set to 16Mhz by   
#define F\_CPU 16000000UL

The main function first initializes the variables, sets the registers for the ADC and puts it in the free running mode. The values set for the ADC registers are as follows:  
1) ADCSRA=7  
2) ADATE,ADEN,ADSC are set to 1.  
3) REFSO is to give the reference and is set to 1.  
4) ADMUX= Channel number  
Ex: ADMUX=0 for taking values from ADC port 0(X direction)  
  
After configuring the ADC , the values from all the channels are stored in variable x,y,z respectively and then the written into the SD Card using the function : write\_data(Filename,Value).  
The code to take values is enclosed in a while loop for taking the readings continuously. There is also a 0.5 second delay in taking each value.  
  
For writing the acceleration values in the micro SD card, I have used pre-existing libraries. I have used the following libraries in my code:  
1) sdcard.c -> To write values in the SD Card  
2) fat.h -> For storing the values in FAT format which can be directly accessed by desktop application

**Learning’s:**The learning’s from the project were as follows:  
1) Learnt a great deal about the Accelerometer sensor, how it works and how to interface with the Microcontroller. Interfacing the sensor with the microcontroller has given me enough confidence to interface any sensor which needs ADC conversion.

2) Understanding how a Capacitor in the Circuit can help reduce noise and give continuous flow of current.

3) I have also learnt how to connect a micro SD card to the micro controller.

4) Learnt how to take value from multiple channels of the ADC at the same time.

5) I also realized the importance of knowing the DC characteristics of the hardware. The accelerometer has 5V written on it but the DC characteristic in the datasheet says that the accelerometer works at voltage of 2 to 3.6 V.

**Critical Challenges**:  
There were many challenges in the project and the project took a lot of time than expected. The challenges were as follows:  
1) Interfacing the Accelerometer took a lot of time . It was not giving correct values .I had to connect a resistance and a capacitor to reduce the noise, current and get the correct value of the ADC.

2) To write the values to the Micro SD card I had to understand all the header files and there implementation. After understanding each function and the semantics of the code, I had to use it on my software.

3) Pin optimizing on the board was also a major challenge.

**References:**1) The Accelerometer datasheet  
2) The headers of the Micro SD card were taken from Abhishek Bhardwaj who took them from an open source website.