DESIGN REPORT

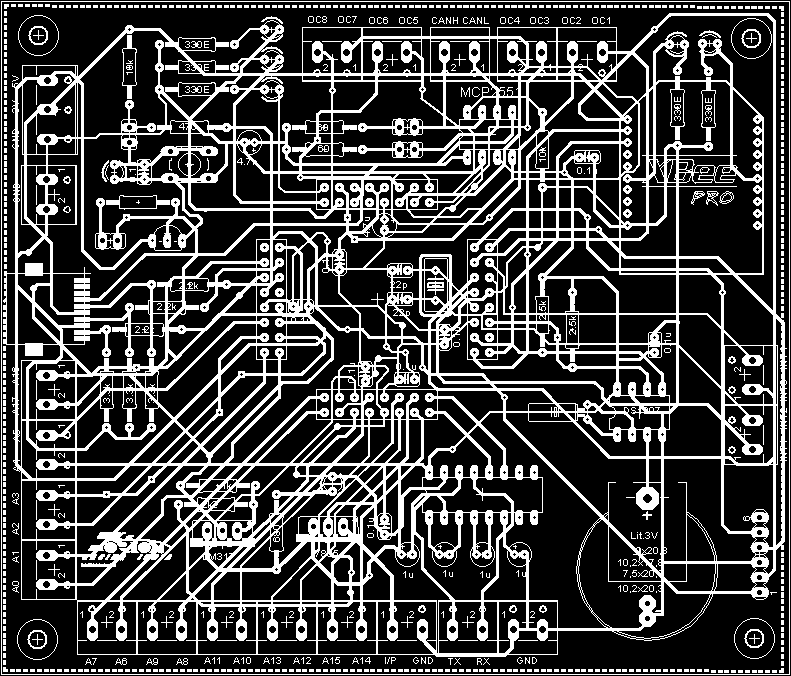
ELECTRONICS

DATA ACQUISITION SYSTEM



At present our data acquisition system uses a DSPIC33FJ128GP706A microcontroller. It has 18 analog channels which take in analog voltages and converts it into digital data and stores it into a USB flash drive

Last year the capacity of our data acquisition system was four channels. This year we modified the code and now the capacity has increased to eighteen channels, allowing us top log eighteen different sensors simultaneously.



Since last year the DAQ had only four channels, the logging rate for the system was 125Hz, which was quite high. But now we have increased the number of channels to 18 thus the logging rate had to be reduced. In order to perfectly log the data we have kept the logging rate to 25Hz. Due to this change our baud rate had to be modified to 38400 from 57600. We are using a peripheral device called VDRIVE to log the data into the flash drive from the microcontroller. VDRIVE acts as the interface between the microcontroller and the flash drive. VDRIVE supports the USB 2.0 protocol.

The casing of our DAQ is too large and has unnecessary space. So this year we decided to change the casing of the DAQ. We have reduced the size of the casing and also changed the colour of the box to black so it would go better with the surroundings. The new case of the DAQ has helped us to save space and also to reduce weight.

Acquisition

1. STS



Shock travel sensor is a linear potentiometer which provides voltage as output

proportional to the position of the slider. When the wheel lifts up from the ground shaft of the sensor tends to move which in turn provides output in terms of voltage. This voltage is sent to the analog channel pin of the micro controller. **Use: Suspension**

1. ACCELEROMETER



Accelerometers are used for measuring the g-force experienced by a body. The accelerometer that we use has the capability to measure all the 3-axis viz x,y,z.

The maximum g-force that can be measured by our accelerometer is 3g which is sufficient for our purpose. This voltage is sent to the analog channel pin of the micro controller.

**Use: Chassis System**

1. Steering Angle Sensor



This is a rotational potentiometer. The output is an analog varying voltage which changes according to the angle of the steering wheel. This voltage is sent to the analog channel pin of the micro controller.

**Use: Steering System**

1. Gear Position Sensor



This is a rotational potentiometer. The output is an analog varying voltage which changes according to the angle of the steering wheel. This voltage is sent to the analog channel pin of the micro controller.

**Use: Engine System**

1. Fuel Level Indicator



This is an angular potentiometer. There is a shaft that varies with the level of the fuel. Corresponding to that a analog voltage is produced. This voltage is sent to the analog channel pin of the micro controller.

**Use: fuel tank**

1. Wheel Speed Sensor



The wheel speed sensor provides us with the information to calculate the wheel speed of the automobile. Our wheel speed sensor works on the Hall effect principle. It provides us with PWM pulses. These pulses can be directly sent to the PWM pins of the microcontroller which can then be logged in the usb flash drive. We can also use a RC network to convert the PWM pulses to analog voltage and then log it in our system.

**Use: Suspension**

1. Break Pressure Sensor



The brake pressure sensor provides the information about the pressure applied on the brake pedal. The output of this sensor is a varying current. We will be using an I to V converter in order to log data from this sensor. This voltage is sent to the analog channel pin of the micro controller.

**Use: Brake System**

Calibration of the sensors was done to ensure proper output. We used MATLAB and LABVIEW to calibrate the sensors. Our microcontroller has a 12-bit ADC. Thus we get a digital output of 0-4095 corresponding to analog voltage of 0-3.3 respectively. We took various reading of the sensor using live telemetry and with the help of LABVIEW. We noted down the readings of the sensor output and plotted the graph in MATLAB. MATLAB was able to provide us with the equation of the graph. Now in our daq code we inserted that equation and hence we were able to take proper reading of the sensor

WIRING

Previously for the relays and the fuses we had separate boxes. This arrangement made troubleshooting difficult for us. To overcome this problem what we have done is that we mounted all the relays on the firewall. This has made assembling the relays and the wiring easier. Also now we have a separate fuse case for every fuse as opposed to last year’s fuse box. This way routing the wires has become easier and also it has helped us to reduce the length of the wires.

Another improvement that we have made is the use of heat shrinks. Last year we had used wire sleeves. The problem with wire sleeves was that due to bending of the wires there was a chance that the wires would snap. The heat shrinks do not pose any problem as such. Another advantage of heat shrinks is that they look good aesthetically, they go with the car.

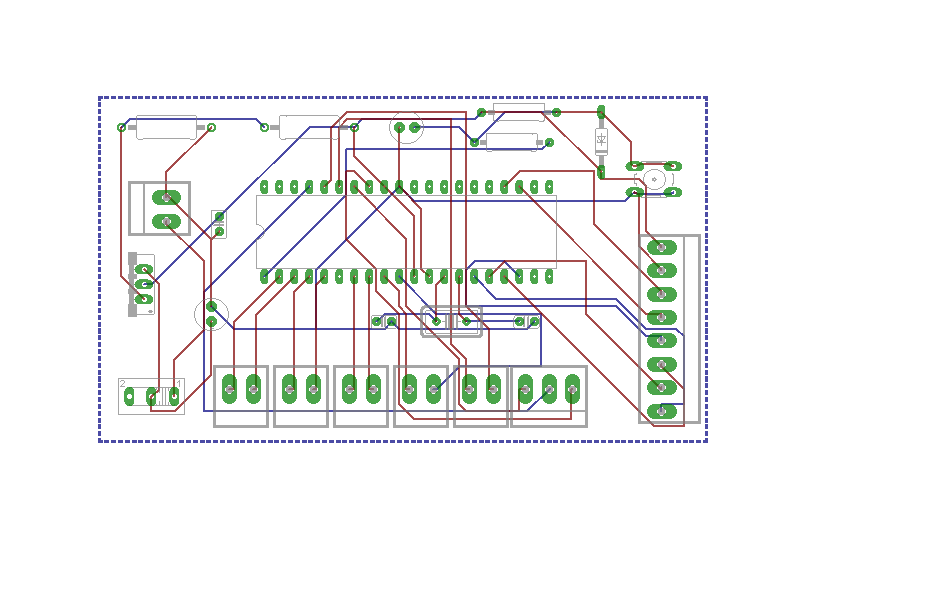
Majority of the wiring has been done using 4mm gauge wire. The starter motor, main kill and fuel pump have been wired using 8mm wire gauge.

The dashboard contains switches for the MOTEC(our ECU), kill, fan, cooling pump, rain-mode and the lcd.

We bought these professional type connectors. But then we realised later that these are not suitable for our task as the crimps were getting loose all the time and we were facing connectivity issues. The crimps would incessantly become loose and would come out of its connector. The new connectors that we bought solved that problem, they are better because the crimps do not get loose and connectivity is maintained during running of the car.

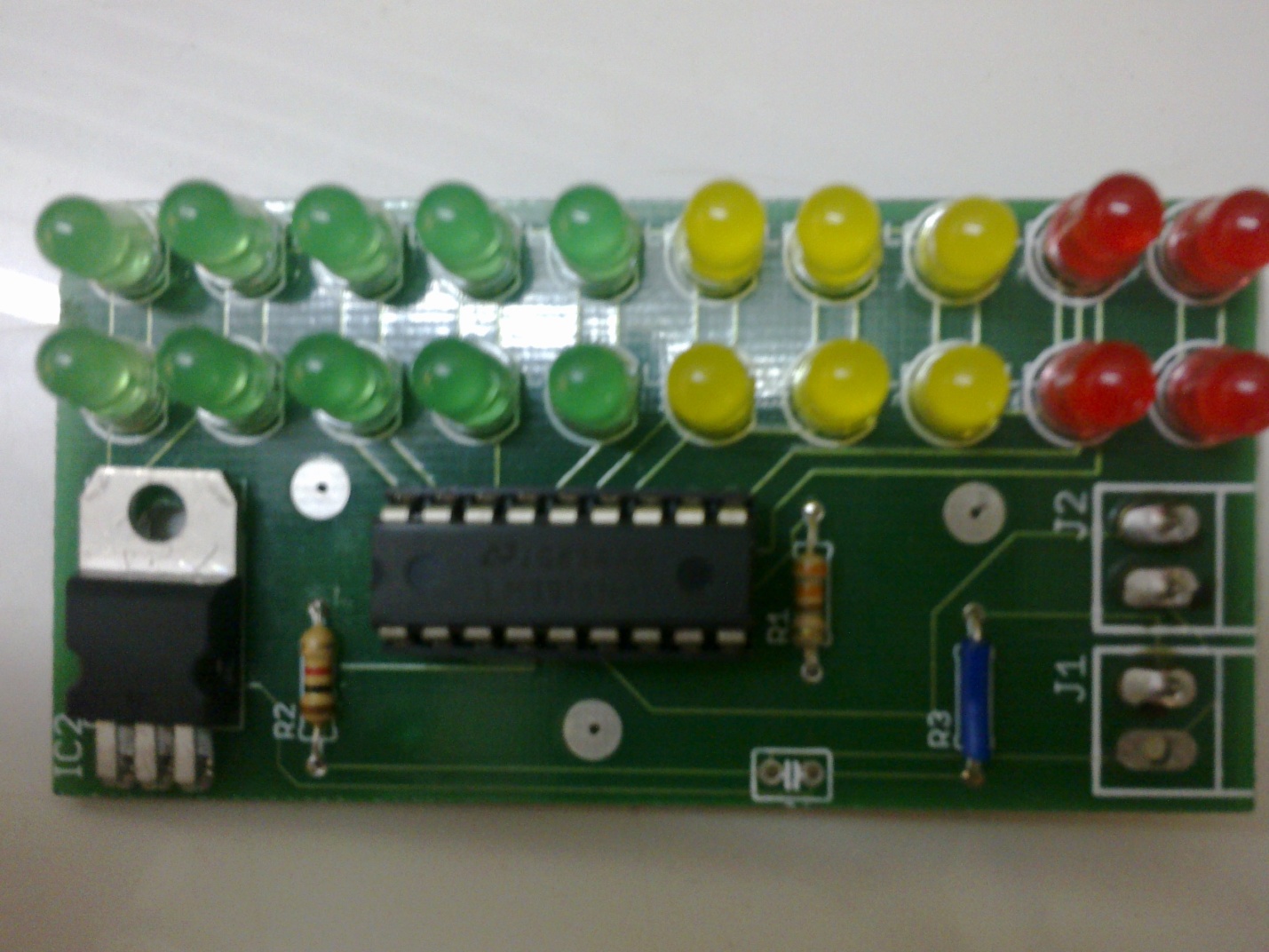
The wiring has also increased as the DAQ has eighteen channels and so all the 18 analog channel needs to be wired with its respective sensor.

DRIVER INTERFACE

LCD on the steering wheel: we are mounting a 20\*4 LCD on the steering wheel to display information to the driver. The information to be displayed includes speed, distance, fuel level, gear position. The LCD is controlled by a PIC18F4520 microcontroller. This controller was chosen because of its ability to get data from 13 different channels using one inbuilt analog to digital convertor. The LCD is controlled in a 4 bit mode for port optimization. Data is taken from the engine as well as the sensors attached on the car. This data is then calibrated and displayed on the LCD. Odometer is also being displayed on the LCD screen and to achieve that we have used the FLASH Memory of the controller to store the odometer reading so that the data is not lost after power off. The LCD code is designed in such a way that a small switch is used to switch the screen from one to another on the same LCD.

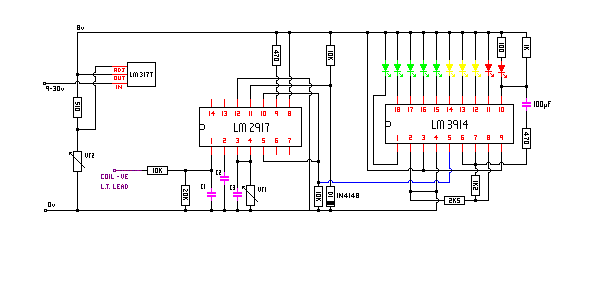


**RPM SHIFT LIGHTS:** This year instead of having two separate circuits for the LED’s we are using a single circuit. This way it saves power and space. The steering wheel does not have enough spacing for a pcb as large as last year’s so we have made a small circuit. We need last year’s circuit photograph for comparison



The circuit consists of LM 3914 which is a bar/dot display driver. A maximum of 10 led’s can be driven. The LED’s are driven according to the voltage variation at the signal pin of the IC which is pin no.5. The signal from the ECU for engine revolution was a varying frequency signal. We thought of using a capacitor with a bleeder resistor but it did not work out. We are now using IC LM 2907 which is a frequency to voltage convertor. This IC converts the frequency from the signal and converts it into analog voltage at 1V/67Hz. LM 7808 is used for regulated voltage supply to the IC.

RPM Shift Lights lets the driver know when to shift gear through the lighting of the led’s giving him a rough idea of the engine rpm.





FUTURE IMPROVEMENTS:

We have decided to use a graphic LCD display for next year for better visual display for the driver while driving the car. Implementation of RPM shifts lights on the LCD controller itself. This can be achieved by reducing the number or channels and making a PORT on the controller available for RPM output.