iCAR+100 : Intelligent Car for medical and police support

***Abstract*— Recent days a major percentage of accidents results in death of passengers due to delay in reaching the medical support. This happens because nearby hospitals did not get accident information in real time. And as always prevention is better than cure. Monitoring the vehicle's performance is first step.**
Here I am trying to implement a prototype for saving human passenger by monitoring the vehicle's performance in real time. In case of accident, this will send the location of car to the centralized mail server. Centralized server will alert the nearest hospital and police station.
I believe that this will reduce the number of accidents and in worst case the death of passengers.

I. Motivation

FARS(Fatality Analysis Reporting System) and NASS(National Automotive Sampling System) CDS(Crashworthiness Data System) data were analyzed to determine time and cause of occupant deaths resulting from light vehicle crashes in early 1990s. The results show 46 percent of the deaths occurred within half an hour, 24 percent between half an hour and an hour and a half and total of 90 percent within 24 hours. Of the deaths occurring during the 1.5 hours following injury, 52 percent were the result of head injuries and 36 percent were the result of thorax injuries.
The fundamental constructs in understanding the causes of deaths is the "three delays" model developed by Thaddeus and Maine. These are

1. Delay in recognizing danger signs/decision to seek care.
2. Delays in reaching a medical facility, and
3. Delay in receiving appropriate care once a facility is reached.

I adopted the "three delays" model as a framework.

II. Objective

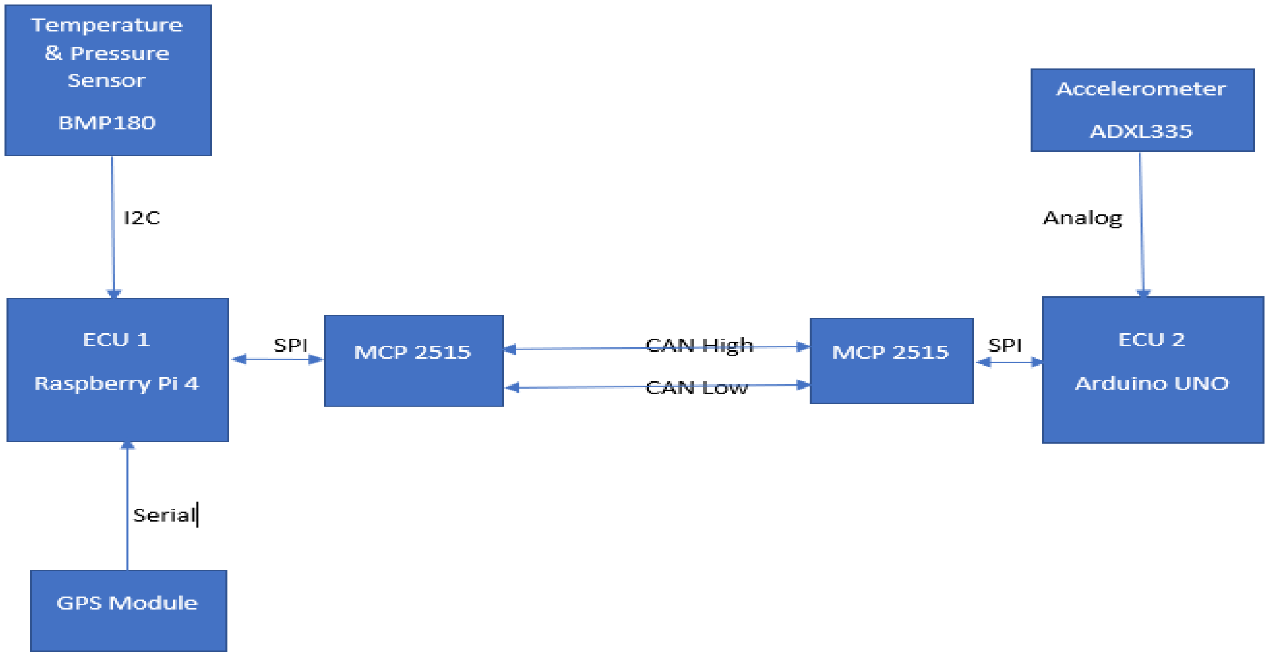
Goal of this prototype is to demonstrate the practical application of real time monitoring of car health which results in saving passenger's lives.

III. Introduction

To rectify the first problem, cars' health monitoring is done.
To rectify the second problem, instant notification that a car crash has occurred, and location coordinated of crash will be sent to centralized server, from there nearby hospital and police stations will be informed.
This advance project brings the possibility of instant notification to pre-hospital care providers that a motor vehicle crash has occurred and need medical support as soon as possible. And monitoring of car health is also is also of great importance for the car owner and car manufacturers.
This project has capability to monitor the car's health in real time.

Two ECUs, accelerometer, temperature and pressure sensor and GPS module are used in the prototype.

IV. Block Diagram

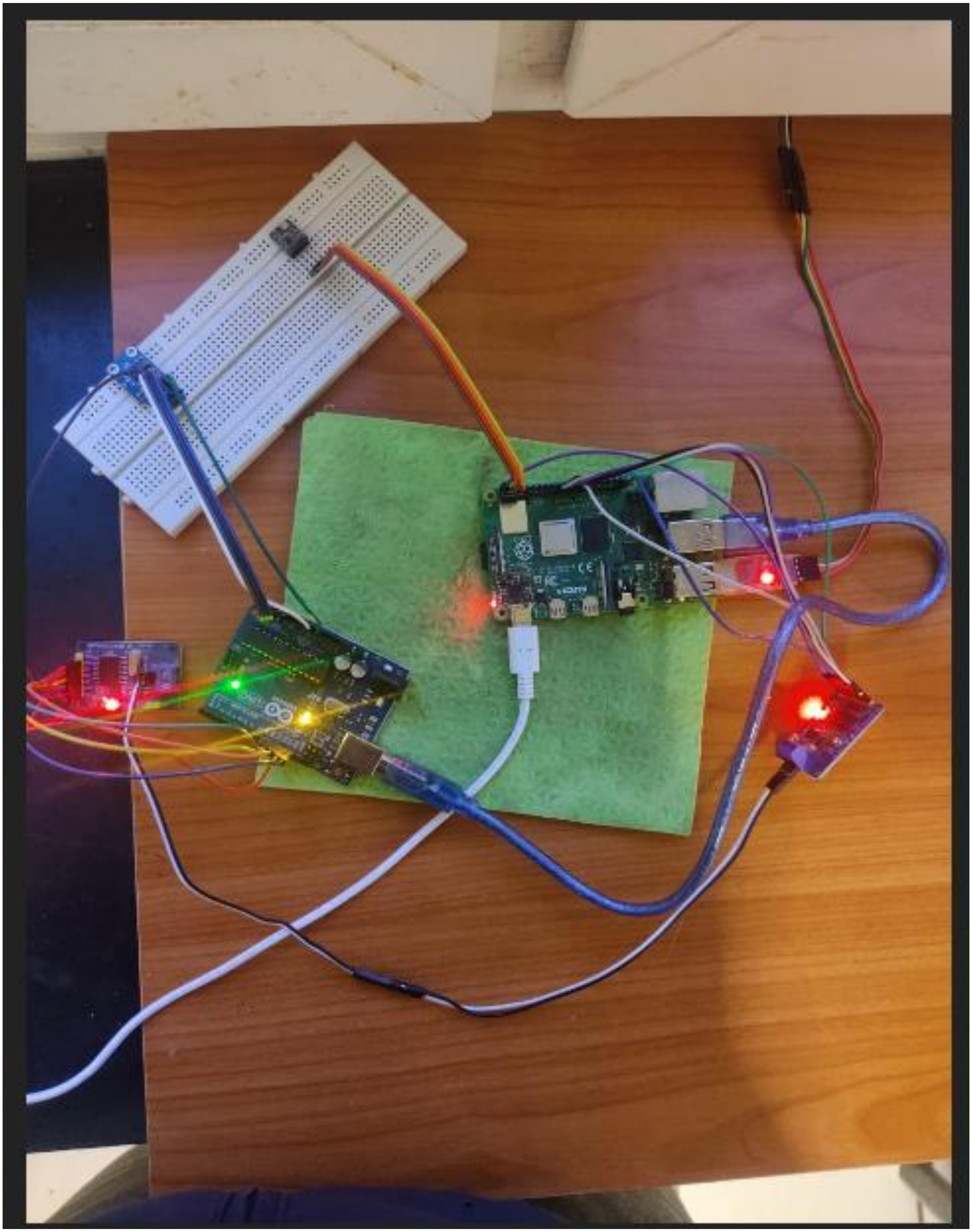


Above figure shows the different blocks of this prototype and its communication protocols used.

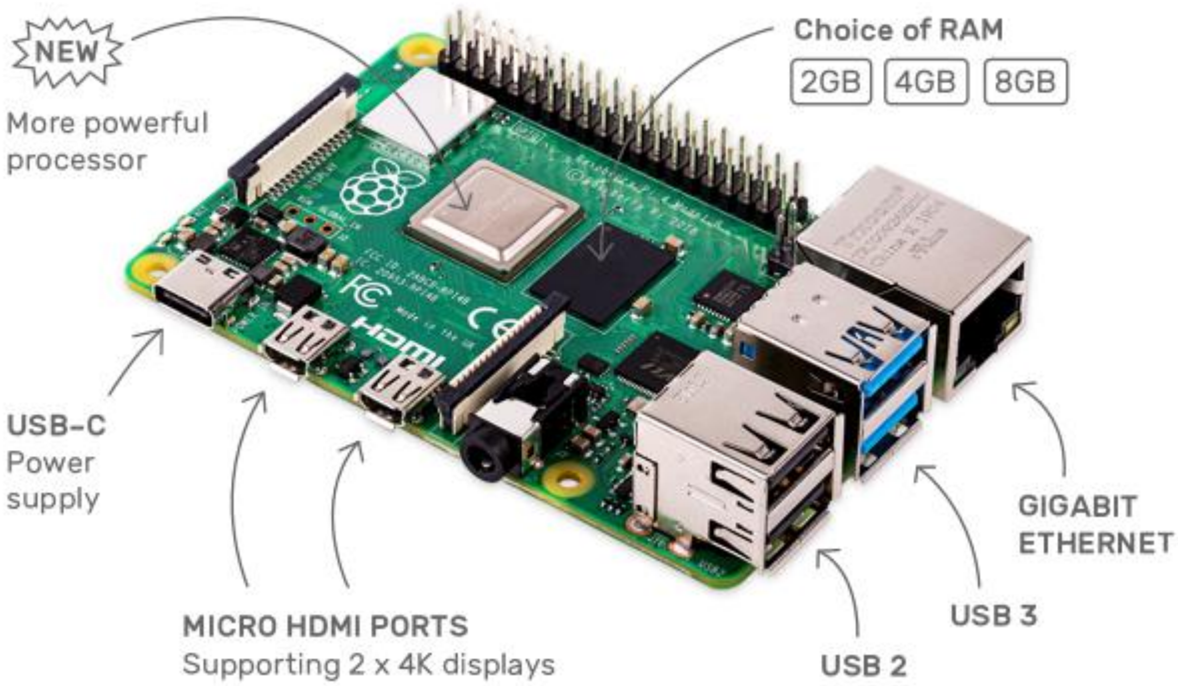
V. Methodology

1) This prototype is developed keeping in mind that less wires are used. So, I used CAN protocol for communication between the different ECUs.
2) Two ECUs are used: Raspberry pi 4 and Arduino uno. Arduino uno is meant to be installed near chassis of car for reading g value in case or accident. Raspberry pi is the main ECU. It relates to temperature & pressure sensor and GPS module.
3) BMP180 Temperature & pressure sensor uses I2C protocol to communicate.
4) GPS module uses serial communication connected through USB.
5) ADXL335 Accelerometer sensor gives analog output which is connected to Arduino because it has ADC.
6) Raspberry pi and Arduino both don’t support CAN, that why MCP2515 is used to convert SPI frame data to CAN frame data.
7) Car health data is collected by raspberry pi and processed using python language and sent to centralized mail server using SMTP (Simple mail transfer protocol)
8) Used SSH to connect and testing the functions of each module.

VI. SYSTEM DESIGN

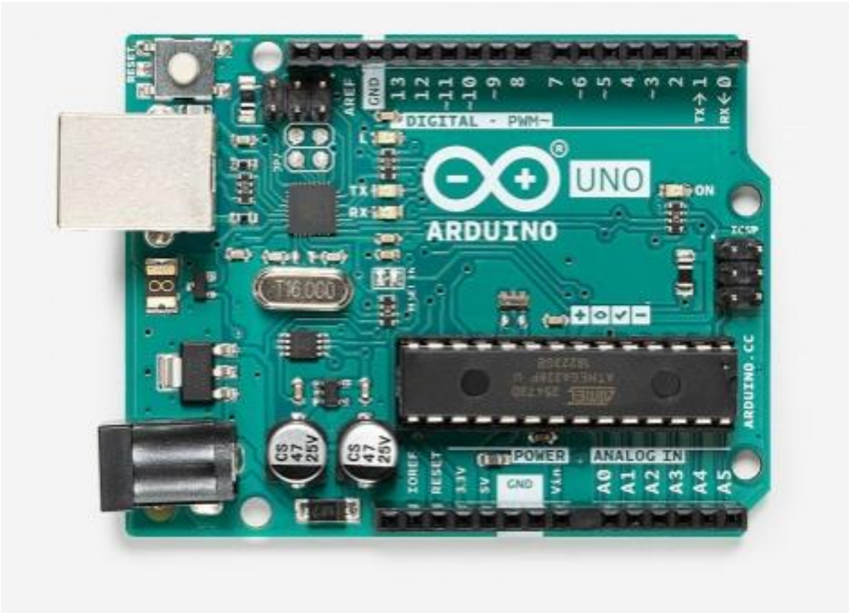


A. Raspberry Pi 4



- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

B. Arduino UNO



Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13

C. Accelerometer : ADXL335

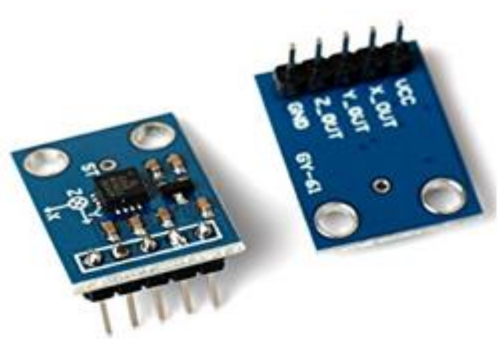


Fig:- ADXL335 Module(Acceleration sensor)

GENERAL DESCRIPTION

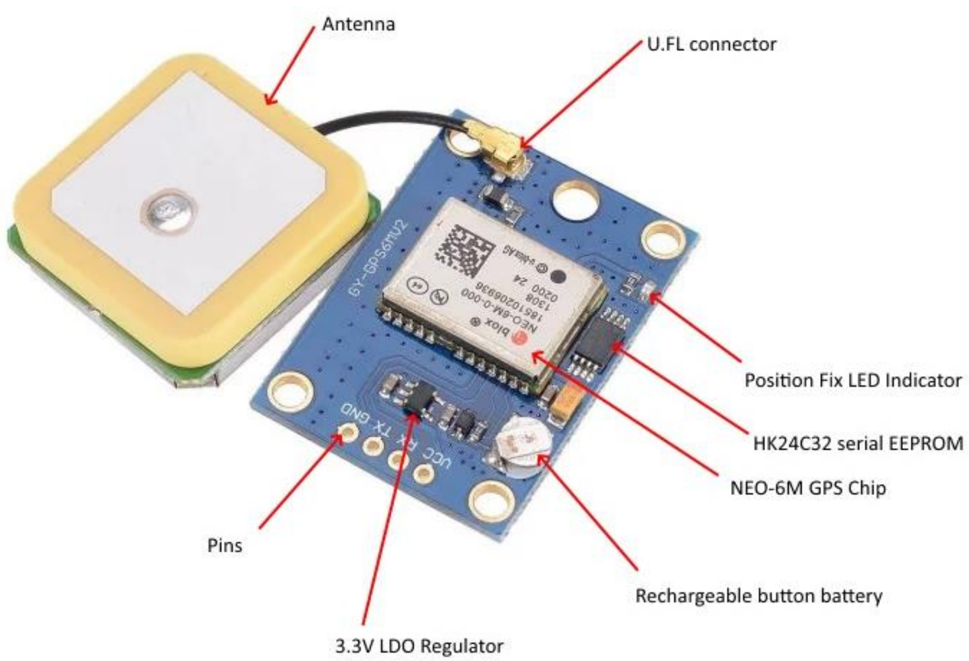
The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis. The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package

compiled of more than one sub-figure presented side-by-side, or stacked. If a multipart figure is made up of multiple figure types (one part is lineart, and another is grayscale or color) the figure should meet the stricter guidelines.

ABSOLUTE MAXIMUM RATINGS

Parameter	
Acceleration (Any Axis, Unpowered)	10,000 g
Acceleration (Any Axis, Powered)	10,000 g
V_S	−0.3 V to +3.6 V
All Other Pins	(COM − 0.3 V) to (VS + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	−55°C to +125°C
Temperature Range (Storage)	−65°C to +150°C

D. GPS Module: NEO-6M GY-GPS6MV2



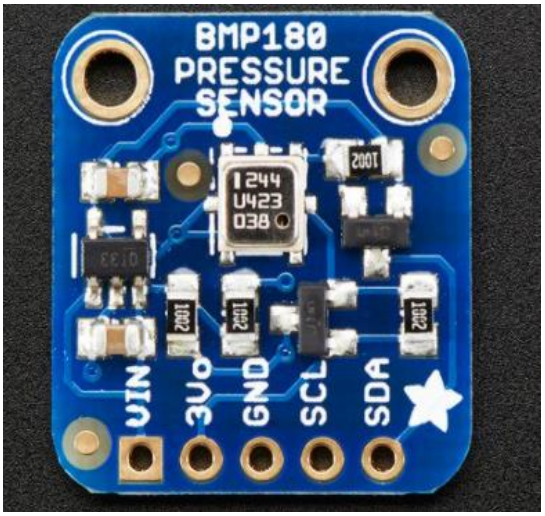
The module is compatible with the NMEA protocol and can be conveniently controlled via the serial interface.

- GPS module with antenna and EEPROM
- compatible with APM2 and APM2.5
- Interface: RS232 TTL
- separate ceramic antenna
- Power: 3-5V
- NMEA protocol
- 9600bps

Scope of delivery:

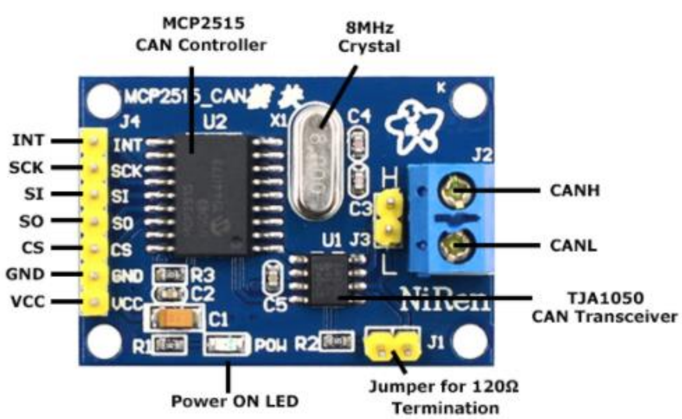
- NEO 6M GPS module
- ceramic antenna

E. Temperature & Pressure Sensor : BMP180



This precision sensor from Bosch is the best low-cost sensing solution for measuring barometric pressure and temperature. Because pressure changes with altitude you can also use it as an altimeter! The sensor is soldered onto a PCB with a 3.3V regulator, I2C level shifter and pull-up resistors on the I2C pins.

F. SPI to CAN Converter: MCP2515



This CAN Bus Module is based on the CAN bus controller MCP2515 and CAN transceiver TJA1050. With this module, you will easily be able to control any CAN Bus device by SPI interface with your MCU.

VII. Conclusion

This System will save the human life by pre-informing the bad health of car to the owner or manufacturers. In worst case when car crashes it will send the crash location to the nearby hospital and police station. By reducing the delay in reaching the medical support to the accident /crash site. Hence, many lives will be saved.

References

Basic Sample code:
https://www.raspberrypi.org/forums/
https://forum.arduino.cc/

Basic GIT Library Used:
https://github.com/adafruit/

Basic Python Learning
https://www.tutorialspoint.com/python/index.htm
: