# Personal Project

Thomas Schenk

Contents

[Personal Project 1](#_Toc129895078)

[Introduction 3](#_Toc129895079)

[Project description 4](#_Toc129895080)

[Requirements of the final product 5](#_Toc129895081)

[Which problems need to be tackled in order to gather data? 6](#_Toc129895082)

## Introduction

This is my personal project for Semester 6. This document will contain everything you need to know about my project, from start till finish. Code that will be written is an exception to this, these files will be put alongside this document as to not clutter. Things like results or graphical images to support the story will be in this document.

## Project description

Being a part of a subgroup which enjoys motorcycles, cars and a lot of other mechanical working things; I gathered quite a bit of knowledge over time about this topic. Knowledge that could be very beneficial to people who don’t have this knowledge. Why? To improve the life of their vehicles.

Lots of people would like to drive older vehicles, but are put off by the seemingly very complicated maintenance which needs to be done on them. While in reality, this maintenance is quite simple. Most of the times, this maintenance however has to be done intuitively, since older vehicles often lack fancy gauges and computers.

My project aims to create such a computer, like a plug and play. While measuring certain diagnostics is a priority; so is saving them. The end user should be able to look back onto their data when they arrive home. That means even though no internet connection is present while driving, the data should be kept and uploaded when there is one.

This product would solve neglect in situations where the owners are not properly informed about maintaining their vehicle.

## Requirements of the final product

This list will serve as requirements for the final product, what is important for the computer to measure? And why? What can we do with the measured data?

* Rpm

With this we can detect how much the user revs the engine, maybe its excessive or maybe its too little. Excessive or improper revving can damage certain parts of the engine in an accelerated way.

* Oil temperature

Engine oil usually has a very high temperature rating, but if it exceeds this rating. Either the driver is driving in a weird/bad way; or some cooling component is damaged. If the oil exceeds its temperature rating in any way, this can impede the effectiveness of this oil. Thus, accelerating wear on the lubricated parts.

* Acceleration/speed

Speed or acceleration is an indicator of how the user drives its vehicle, speed has so much influence on all factors in a vehicle; that data can easily be derived from it.

The hardware the final product will be using will all be used on an ESP32. This choice is specifically made since it integrates with various communication methods from factory. But it also supports low battery usage which could be great for use in the future.

## Software diagram

## Diagram Description automatically generated

## Which problems need to be tackled in order to gather data?

Currently there are some problems which need to be tackled in order to gather the data previously mentioned.

* How are we going to measure RPM?

Most of the time RPM is already integrated in a vehicles dashboard. But I want to tailor this product more towards scenarios where this is not the case; how do we determine engine RPM when this is not integrated?

Usually engine RPM meters are mechanically implemented. This is because to measure the RPM of an engine, you need access to the crankshaft. Mechanically this is way easier to achieve than electrically. However both need to be implemented by design, it is near impossible to implement a mechanical or electrical solution based on the crankshaft.

That is why we are going to implement a system that makes use of a trivial part of the workings of an engine; the spark plugs. By determining the firing pattern, we can approach the number of times the engine fires over; and thus determine how fast the engine is spinning.

* How are we going to measure oil temperature?

In all applications, an oil temperature reading is going to need specialized hardware. Especially if this hardware is missing from the vehicle in the first place.

Most of these oil sensors from OEM manufacturers rely on a thermistor which determines the resistance over the sensor. Knowing this, we can solve our missing sensor problem by making sure we implement a feature where a temperature sensor can be added.

* How are we going to measure speed/acceleration?

Speed measurements often rely on GPS. This is because to measure speed, a current and previous location are needed in order to calculate the speed.

Accelerometers, a group of sensors which are also involved in movement calculations, lack these datapoints. Which would require making a map taking the starting point into account and calculate velocity between the travelled distances. This however would take an enormous amount of calculations and most of all, would require a lot of storage to accommodate for the ever growing map that will eventually be created.

This is a shame, since acceleration isn’t a native measurement of a GPS system. This would mean that for acceleration, we need to add another sensor.   
However, the prognosis is that we can estimate this using the distance and speed gathered by the GPS.

## Used hardware

Computational device:

* ESP32

Why:  
ESP32 is a platform which natively houses a lot of functionality already. Unlike an Arduino, ESP32 natively supports wireless communication methods such as radio, Bluetooth and WIFI. Most of these sensors can be expanded upon by introducing very little hardware. LORA for example.

Sensors:

* GPS-unit
* RPM-Circuit
* Temperature circuit