SPTD Speech

# 1: Benvenuti

Buongiorno, stimati membri della commissione, professori, ingegneri e spettatori.

Innanzitutto, vorrei ringraziarvi per essere qui, mentre presento la mia tesi finale, per poter conseguire il titolo di Laurea in Ingegneria Informatica.

Prima di addentrarci in quella che è la presentazione di questa smart-plug top Digital, in particolare del firmware che vi risiede, sarà necessario introdurre il campo di applicazione del prodotto.

A seguire troviamo una panoramica dell’ambiente di sviluppo, della strumentazione utilizzata per compilare, debuggare e infine programmare il codice all’ interno del microcontrollore.

Infine, viene illustrata l’architettura del codice nonché i vari moduli realizzati, con una particolare nota del protocollo di comunicazione implementato, quella della LIN.

Una Premessa prima di iniziare: Poiché la tesi è in lingua inglese, per mantenere un certo grado di coerenza, si è optato di mantenere anche la presentazione in inglese.

(Pausa Ripigliati)

{7-8 minutes left}.

# 2: Intro to Ignition Automotive also known as Powertrain

Let us start by analyzing this visual representation of the circuitry involved in this controlled ignition system.

This process involves a series of components, working in harmony, to ignite the air-fuel mixture within the combustion chamber.

The primary source of energy is the battery, providing a 14 V nominal input.

This current flow is then taken to the ignition coil, when the ignition switch, the gatekeeper of this process, is switched on.

The ignition coil is responsible for not only generating the high voltage required for the ignition, but also for storing this energy.

This mechanism that generates such voltage is controlled by the opening and closing of the contact breaker, also known as points, upon reaching precise thresholds.

This high voltage tension is then transferred from the distributor housing to the distributor cap, through the spinning of the rotor. These components allow the correct distribution of this tension, enabling them to reach the spark plugs, where the ignition will take place.

The spark plug, will then ignite the air-fuel mixture, initiating the combustion and setting the engine in motion.

This cycle is repeated thousands of times, per second, in order to propel our vehicles.

Moving on, let’s further analyze the main components in this system (Ignition coil and spark plugs), that our product intends to diagnose.

# 3: Ignition Coil

First, we have the Ignition Coil, a very critical component.

The ignition coil is typically composed of 2 coils, primary and secondary, along with a transformer.

The primary coil serves as the initial recipient of the current coming from the battery, whose flow is controlled by a resistor like the one shown in the previous slide.

It stores this energy, building up a magnetic field within its windings (avvolgimenti).

When the current flowing through the primary coil is interrupted, the collapsing magnetic field induces a high voltage pulse in the secondary coil.

The secondary coil, wound around the primary coil, receives this high voltage pulse, and transforms it into an even higher voltage output.

Essentially, the ignition coil acts as an energy reservoir, accumulating and then discharging energy onto the secondary coil. It serves as the vital link between the low voltage supplied by the battery and the high voltage required for spark generation.

# 4: Spark Plug

Moving on, the next main component in this system is the spark plug. The module is responsible for igniting the air-fuel mixture.

The central electrode, typically made of durable materials like nickel alloy or platinum, extends into the combustion chamber. Surrounding the central electrode is the ground electrode, creating a small gap known as the spark plug gap.

When the ignition system delivers a high voltage pulse, the spark plug transforms this electrical energy into a powerful electrical arc. This arc across the spark plug gap, ignites the air-fuel mixture.

This controlled explosion generates the force needed to drive the engine's pistons and propel the vehicle forward.

These components are fundamental in initiating the controlled explosions that power our engines.

# 5: SPTD

However, the lack of diagnostic information fro b

m these crucial components affects our ability to monitor and optimize the combustion process effectively.

It is in this context that the SPTD (Smart Plug-Top Digital) system emerges as a groundbreaking solution. The SPTD system aims to address the diagnostic limitations by integrating advanced technology.

This innovative system enables real-time data collection, analysis, and reporting of critical parameters such as spark intensity, spark timing, and combustion efficiency.

With the SPTD system, automotive engineers and technicians gain valuable insights into the performance of the ignition system, allowing for early detection of potential problems.

The requirements necessary to achieve our goal of integrating the SPTD system into the ignition process include:

First and foremost, a robust microcontroller environment. The selection of an appropriate microcontroller is vital to ensure the smooth operation of the SPTD system.

Additionally, we must utilize the necessary modules within the microcontroller to execute the required functions, modules including analog-to-digital converters (ADC), pulse-width modulation (PWM) controllers, and timers, among others.

Furthermore, it is imperative to correctly interpret the various input and output (I/O) signals within the microcontroller, essential for monitoring and controlling the ignition process, and precise interpretation is critical for accurate operation.

In addition to internal communication within the microcontroller, our system must also establish communication with external devices, specifically the Engine Control Unit (ECU). This communication is facilitated through the Local Interconnect Network (LIN) protocol, a widely used standard in the automotive industry.

# 6: Firmware Environment and tools

The firmware development process is facilitated by the MPLAB X IDE, a robust and feature-rich Integrated Development Environment.

In conjunction with the MPLAB X IDE, we leverage the power of MCC (Microchip Code Configurator), an automated code generation tool. MCC simplifies the development process by generating code snippets and libraries based on the configuration settings and requirements of our project. This automation significantly reduces development time and ensures code consistency and reliability.

The firmware for the SPTD system is predominantly written in the C programming language, known for its efficiency and versatility.

To compile the C code and generate executable firmware, we utilize the XC16 compiler, specifically designed for Microchip microcontrollers.

When it comes to programming and debugging the firmware, we rely on the PICkit™ On-Board (PKOB4). An integrated tool that allows for real-time debugging, enabling us to identify and rectify any issues or anomalies within the firmware.

In addition to the programming and debugging tools, we employ the Peak Systems PLIN-USB for testing purposes. This specialized tool enables us to evaluate the communication and functionality of the LIN protocol, ensuring seamless interaction between the SPTD system and external devices such as the ECU.

# 7: dsPIC33CDVL64MC106

As we shift our attention to slide 8, we come face-to-face with the heart of our SPTD system—the chosen microcontroller. This slide highlights the key features and capabilities of the dsPIC33cdvl64MC106 microcontroller, which serves as the driving force behind our innovative ignition solution.

The dsPIC33 is a powerful 16-bit device with integrated digital signal controller capabilities. A unique combination that empowers our SPTD system to efficiently process and manipulate signals, enabling precise control and optimization of the ignition process.

To facilitate interrupt handling and response, the dsPIC33cdvl microcontroller employs an Interrupt Vector Table. This table organizes and manages the various interrupt sources, allowing the microcontroller to swiftly respond to critical events and execute the corresponding interrupt service routines.

One standout feature of the chosen microcontroller is the integrated LIN transceiver, specifically the ATA663211. This built-in transceiver simplifies the implementation of the LIN protocol, eliminating the need for additional external components.

In terms of memory capacity, the dsPIC33cdvl microcontroller has 56 KBytes of flash memory providing ample space for storing the data and other essential information.

Cost-effectiveness is another significant advantage of the chosen microcontroller, Despite its advanced features.

# 8: FSM

This slide provides an overview of the firmware's design and organization, emphasizing its interrupt-driven architecture and the three key modules that contribute to its seamless operation.

To facilitate independent development and testing, the firmware employs a modular code structure. This modular approach partitions the codebase into distinct and autonomous modules, enabling separate development, testing, and maintenance of each component.

As we can see in the FSM that represents the architecture, the code flow follows a very simple and straightforward state transitioning diagram.

If no interrupt is detected we are in the IDLE state.

Depending on the Interrupt received, the interrupt service routine will call the proper module, following the assigned priority in the vector table.

Let us now focus on the three fundamental modules that constitute the core of our SPTD system:

# 9: Trigger Handler

Trigger Handler:

The Trigger Handler module is responsible for monitoring the ignition trigger signals and initiating the ignition sequence. It detects the necessary trigger events, such as the engine's rotational position, and triggers the corresponding actions within the firmware. By accurately synchronizing the ignition process with the engine's dynamics, the Trigger Handler module ensures optimal ignition timing and performance.

# 10: Data Handler

Data Handler:

The Data Handler module plays a pivotal role in managing and processing crucial data within the firmware. It collects, validates, and organizes various sensor data, engine parameters, and system variables necessary for precise control and monitoring. The Data Handler module utilizes efficient algorithms to facilitate real-time data processing, enabling accurate ignition control and diagnostics.

# 11: LIN

# 12: LIN Handler

LIN Handler:

The LIN Handler module establishes and manages communication with external devices, particularly the Engine Control Unit (ECU), through the LIN (Local Interconnect Network) protocol. It handles the transmission and reception of LIN messages, allowing seamless data exchange and synchronization between the SPTD system and the ECU.

# 13: Thanks for attending

I would like to cherish this opportunity to thank my company tutor, esteemed Engineer Simone Daniele, who followed me in this firmware development process as well as supporting me in my role as Firmware and Software Engineer, in the company.