UNIVERSIDADE TECNOLOGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E POS GRADUAÇÃO
DEPARTAMENTO ACADEMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇAO EM SISTEMAS EMBARCADOS PARA A INDUSTRIA AUTOMOTIVA

MARCELO DOS REIS MANSANO

PLATAFORMA IOV UTILIZANDO AGL: desenvolvimento e aplicação

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2017

MARCELO DOS REIS MANSANO

PLATAFORMA IOV UTILIZANDO AGL: desenvolvimento e aplicação

Monografia de Especialização, apresentado ao Curso de Especialização em Sistemas Embarcados para a Industria Automotiva, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Luiz Copetti

CURITIBA

2017

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

DIRPPG
DAELN
CESEB

TERMO DE APROVAÇÃO

PLATAFORMA IOV UTILIZANDO AGL: desenvolvimento e aplicação

por

MARCELO DOS REIS MANSANO

Esta Monografia foi apresentada em 30 de novembro de 2017 como requisito parcial para a obtenção do título de Especialista em Sistemas Embarcados para a Industria Automotiva. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____

Prof. MSc. Luiz Copetti
Orientador

_____

Prof. MSc. Omero Francisco. Bertol
Orientador

_____

Prof. Dr. Kleber Kendy Horikawa Nabas
Coordenador do Curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho à minha esposa Letícia e meu filho Eric.

RESUMO

MANSANO, Marcelo. **Plataforma IoV utilizando AGL:** desenvolvimento e aplicação. 2017. 29 f. Monografia De Especialização – Curso De Especialização Em Sistemas Embarcados Para A Industria Automotiva, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

A popularização da Internet das Coisas abriu caminhos para aplicações na indústria automotiva. A utilização de conceitos de comunicação em nuvem, agregação de dados e processamento remoto incentivaram grandes marcas do mundo automotivo ´a investir em padrões e tecnologias. Com isso, a *Linux Foundation* criou o Automotive Grade Linux, um sistema base para o desenvolvimento base de aplicações de informação e entretenimento automotivo. Este trabalho aplicou o AGL em um conjunto *Raspberry Pi* 3, um módulo de comunicação CAN e um celular para comunicação com a nuvem, para propor um framework base para projetos de *Internet of Vehicles*, utilizando apenas de ferramentas de software livre. Uma aplicação de testes foi feita para comprovar a capacidade da plataforma e do framework, realizando a captura de dados e envio para um servidor central através de uma rede 4G. Os resultados provaram a capacidade do framework para implementação de aplicações de captura e processamento de dados remotos. Concluiu-se que o AGL e o framework proposto podem ser utilizados como base em projetos de *Internet of Vehicles*.

Palavras-chave: Internet of Vehicles. Linux. AGL. Raspberry Pi. CAN. 4G.

ABSTRACT

MANSANO, Marcelo. **IoV Platform using AGL:** Development and Application. 2017. 29 f. Monografia De Especialização – Curso De Especialização Em Sistemas Embarcados Para A Industria Automotiva, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

The popularization of the Internet of Things paved the way for applications in the automotive industry. The use of concepts of cloud communication, data aggregation, and remote processing have encouraged major brands in the automotive world to invest in standards and technologies. With this, the Linux Foundation created the Automotive Grade Linux, a base system for development automotive information and entertainment applications. This work applied the AGL in a Raspberry Pi 3 with a communication module CAN and a smartphone for communication with the cloud, to to propose a base framework for Internet of Vehicles projects, using only open-source software tools. A test application was made to prove the capability of the platform and of the framework, performing data capture and sending to a central server through a 4G network. The results proved the capability of the framework to help implement capture and process remote data applications. It was concluded that the AGL and the proposed framework can be used as a basis for Internet of Vehicles projects.

Keywords: Internet of Vehicles. Linux. AGL. Raspberry Pi. CAN. 4G.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IoT	Internet of Things
IoV	Internet of Vehicles
ECU	Engine Control Unit
UCB	Unified Control Base
CMCU	Centralized Management and Control Unit
ITS	Intelligent Traffic Systems
IVI	In-Vehicle Infotainment
CAN	Control Area Network
LIN	Local Interconnect Network
ADAS	Advanced driver-assistance systems
MOST	Media Oriented Systems Transport
GPS	Global Positioning System
AMB	Automotive Message Broker
QEMU	Quick Emulator

SUMÁRIO

1 INTRODUÇÃO

A Internet das Coisas (IoT) promete conectar entre 50 a 100 bilhões de dispositivos até 2020 (SUNDMAEKER et al., 2010), dentre estes os veículos automotores se destacam pela riqueza de informações disponíveis, que podem ser obtidos do diretamente dos sensores do veículo, dos passageiros e da estrada. (GERLA et al., 2014) define um termo para a aplicação de IoT em veículos como *Internet of Vehicles* (IoV). Hoje o IoV já está presente nos veículos através dos sistemas de *In-Vehicle Infoteinment* (IVI), onde os passageiros têm informações e controles sobre os sensores através de aplicativos de celular como o sistema mySPIN® da Bosch®.

Com o IoV é possível expandir as formas de controle e acesso aos dados do veículo, de forma a exportar para o usuário interfaces de acesso através de protocolos de populares, tais como Bluetooth® ou Ethernet® (GERLA et al., 2014). Desta forma, é possível manter o veículo conectado a um servidor onde o usuário poderá manter histórico dos dados, obter informações atuais e estatísticas, além de possibilitar o controle remoto de diversas funcionalidades do veículo, conectando o hardware de IoV diretamente a ECU do veículo (GAI; VIOLANTE, 2016).

Outro caso de uso para esta aplicação e em cidades inteligentes, onde carros podem se cooperar de forma ativa através da troca de dados para identificar problemas, informar de perigos ou até mesmo possibilitar a melhor interação entre carros autônomos (GERLA et al., 2014). Este tipo de aplicação mostra como o uso de IoV depende de normas de segurança bem definidas para acesso e troca de informações entre pontos.

Para atender estas expectativas, a *Linux Foundation*, em parceria com diversas montadoras de veículos automotores, vem desenvolvendo nos últimos anos um projeto para adaptar sua base de código para as necessidades impostas pelo mercado automobilístico (COUNTS, 2017) e complementando padrões já existentes como o AUTOSAR (GAI; VIOLANTE, 2016).

Para este projeto foi dado o nome de *Automotive Grade Linux* (ou AGL). Como este projeto é realizado em forma de código livre, torna-se possível a qualquer pessoa desenvolver aplicações de IoV utilizando-se do padrão *Unified Code Base* (UCB) (KERNER, 2016).

Este trabalho descreverá uma plataforma criada para o desenvolvimento de aplicações de IoV utilizando AGL, utilizando como exemplo uma aplicação para capturas de dados de um veículo de forma remota, através de uma *Raspberry Pi* 3. Neste trabalho, o hardware AGL estará conectado a um servidor de dados através da rede 4G, e disponibilizara uma interface ethernet para acesso local, via SSH. São utilizadas ferramentas de código livre e hardwares populares para a implementação de um protótipo para esta aplicação. Por fim, serão discutidos os resultados obtidos e serão propostos trabalhos futuros para o desenvolvimento da plataforma.

1.1 PROBLEMA

O conceito de IoV e o desenvolvimento do AGL são tópicos muito recentes (KLAVMARK; VIKINGSSON, 2015). Por isso poucos trabalhos foram publicados e poucas implementações foram propostas até então. Empresas como HARMAN®, Intel® , Jaguar®, Land Rover® , Nissan® , Samsung® e Toyota® juntaram-se a iniciativa proposta pela *Linux Foundation*, alavancando $ 10 Bilhões de investimento ao projeto (AGL, 2012). Recentemente a Toyota anunciou que o modelo 2018 do *Toyota Camry*® será o primeiro veículo comercial a trazer embarcado AGL para IVI (TOYOTA, 2017), dando um importante passo para a popularização da plataforma em relação aos seus concorrentes (como o projeto GENIVI e o Automotive Grade Android) (NEWCOMB, 2012).

Dado este contexto, propõe-se aqui implementar uma plataforma de desenvolvimento para IoV baseada em AGL, com código livre, seguindo as especificações apresentadas pela *Linux Foundation* e o código base UCB 2.0, visando obter uma base de desenvolvimento para novos projetos e trabalhos futuros. A aplicação exemplo é proposta para avaliar a aplicabilidade de um sistema IoV com AGL, afim de obter-se uma funcionalidade que possa ser utilizada em veículos automotores para detecção de anomalias, observando diversos parâmetros do mesmo.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

1.2.1 Objetivo Geral

Demonstrar a aplicabilidade de AGL e IoT no mercado automobilístico, verificando a importância dos padrões de segurança e usabilidade definidos pela Linux Foundation, através de uma plataforma constituída de software e hardware com diferentes tipos de interface de comunicação entre o veículo, o usuário e a internet.

1.2.2 Objetivos Específicos

- Desenvolver uma plataforma AGL utilizando ferramentas de código livre em hardwares de fácil acesso, seguindo o UCB 2.0.
- Definir um framework base para o desenvolvimento de aplicações IoV utilizando AGL.
- Analisar a capacidade do AGL para IoV, observando quesitos de funcionalidade, segurança, facilidade de utilização e perspectivas.
- Verificar a funcionalidade da plataforma com a aplicação de armazenamento e detecção de anomalias em dados remotos do veículo.
- Disponibilizar a plataforma desenvolvida para trabalhos futuros.

1.3 JUSTIFICATIVA

Visto a atual popularização do tema de IoT faz-se importante a demonstração de como este pode ser aplicado ao mercado automobilístico, expandindo as funcionalidades já existentes nos veículos. Quanto mais popular este tema se torna, maior será a necessidade de se ter uma plataforma base para o desenvolvimento de novas aplicações. Com isto, este trabalho visa abrir caminho pra novos desenvolvimentos, facilitando assim o início de novos projetos. A implementação da aplicação de exemplo visa solucionar um problema real. Utilizando-se de conceitos de IoV e do AGL poderá demonstrar-se a importância da implementação da plataforma proposta e das ferramentas já disponíveis para o futuro da tecnologia automotiva, esperando-se inspirar novos estudos e projetos na área.

1.4 ESTRUTURA DO TRABALHO

O trabalho terá a estrutura baixo apresentada.

- **Capítulo 1 – Introdução**: serão apresentados o tema, o problema, os objetivos da pesquisa, a justificativa e a estrutura geral do trabalho.
- **Capítulo 2 – Fundamentação Teórica**: serão apresentadas informações sobre *Internet of Vehicles, In-Vehicle Infotainment, Automotive Grade Linux* e sobre o *Yocto Project*.
- **Capítulo 3 – Desenvolvimento do Tema**: serão apresentados os detalhes de implementação do hardware, da construção da imagem de testes, do framework proposto e da funcionalidade de detecção de anomalias.
- **Capítulo 4 – Apresentação e Análise dos Resultados**: neste capítulo serão mostrados os resultados obtidos através de testes da funcionalidade
- **Capítulo 5 – Considerações finais**: serão retomados a pergunta de pesquisa e os seus objetivos. Haverá a indicação de como foram solucionados, respondidos, atingidos, por meio do trabalho realizado. Além disto, serão sugeridos trabalhos futuros que poderiam ser realizados a partir do estudo realizado.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo irá descrever brevemente conceitos utilizados no desenvolvimento deste trabalho, introduzindo e definindo terminologias e conceitos utilizados neste trabalho.

2.1 INTERNET OF VEHICLES

O termo *Internet of Vehicles* foi definido em (GERLA et al., 2014) para especificar as aplicações de IoT desenvolvidas para a indústria automotiva. As principais aplicações para ela giram em torno do ITS (*Intelligent Traffic Systems*) e IVI (*In-Vehicle Infotainment*), onde a troca de informação entre dispositivos, veículos, humanos e unidades de trânsito dependem de comunicação, armazenamento e processamento eficiente de dados.

O IoV permite a utilização de muitos tipos de comunicação sem fio, como WLANs, WiMAX, redes de celular e satélite para que a segurança, qualidade e facilidade de transferência de dados seja atingida. Por questões de segurança, (FANGCHUN et al., 2014) propõe que estas comunicações sejam gerenciadas por uma central de controle e gerenciamento no veículo, chamada de CMCU (*Centralized Management and Control Unit*). Este dispositivo deve ser capaz de se comunicar com outros CMCUs e dispositivos em nuvem para troca e processamento de informações de forma segura e com alta disponibilidade.

Dentro dos veículos a CMCU pode acessar dados e informações em tempo real dos sensores e dispositivos de *infoteinment* através de redes já utilizadas pela indústria, como CAN, LIN, ADAS e MOST. Desta forma os dados podem ser processados localmente, como em (WAN et al., 2016) ou armazenados e processados externamente assim como feito por (LOWREY et al., 2003).

A Figura 1 mostra o modelo definido por (FANGCHUN et al., 2014) para uma rede IoV baseada em multi-protocolo. O conjunto de veículos, aparelhos e humanos dentro deste ambiente pode ser chamado de *Swarm* (do inglês enxame). As redes intra-veiculares possibilitam a interface, ação e troca de informações entre os atores do sistema interno (sensores, humano e sistemas de *infoteinment*).

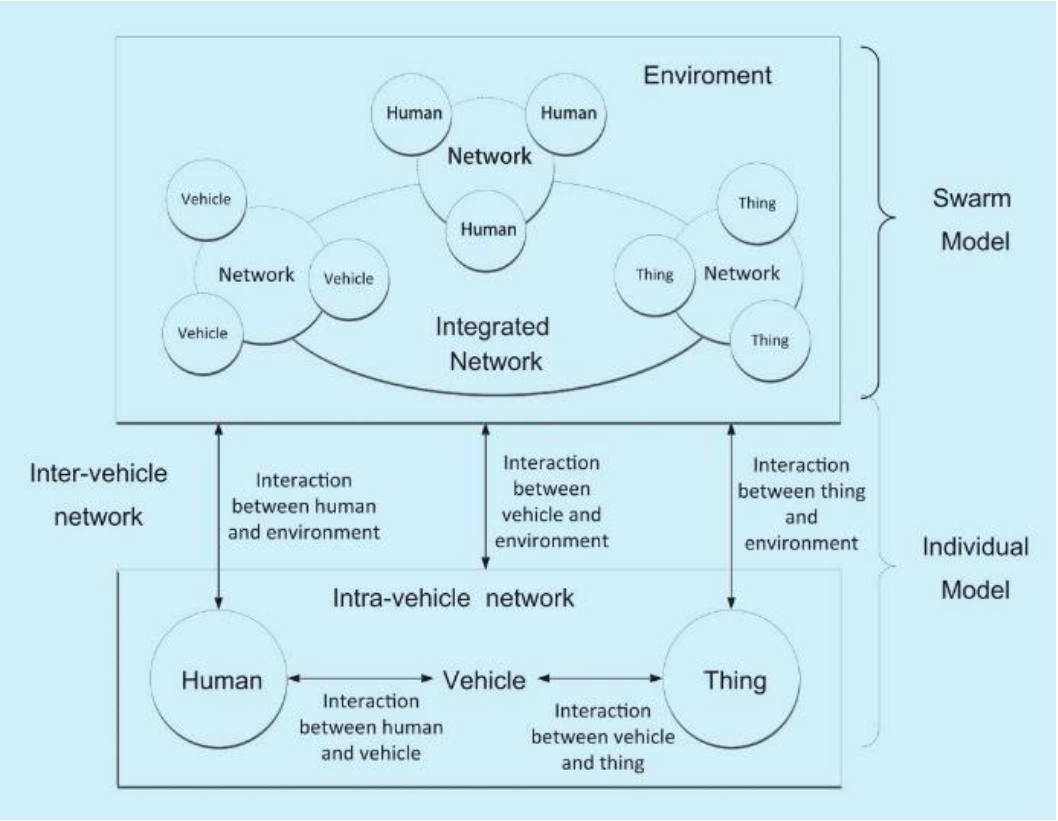

Figura 1 - Modelo definido como base para aplicações IoV.

Fonte: Fangchun et al. (2014)

2.2 IN-VEHICLE INFOTAINMENT

In-Vehicle Infotainment (IVI) é o termo utilizado pela indústria automotiva para se referir aos sistemas que combinem a entrega de informações e entretenimento aos passageiros de um veículo (KLAVMARK; VIKINGSSON, 2015). Para isto, sistemas de IVI devem conter interfaces humano-máquina (como telas, controles e alto-falantes) e conexão, tanto intra-veiculares como externas. Algumas das aplicações de IVI incluem: navegação por GPS®, central multimídia e diagnóstico do veículo em tempo real.

Com o surgimento de novas tecnologias de comunicação internas, como redes de alta-velocidade para transmissão como a MOST®, e externas, como Bluetooth® ou 4G/5G para outros dispositivos, possibilitaram a expansão desta tecnologia. Trabalhos como (BELYAEV et al., 2014) mostram como a evolução das tecnologias de comunicação ajudam motoristas e passageiros a interagir com o veículo.

Estes sistemas têm se tornado um grande diferencial na indústria automobilística pois permitem as montadoras inovar e criar novas tecnologias que

possibilitam mais conforto e segurança aos passageiros e contribuem para um melhor ambiente de tráfego. Empresas, como a Volkswagen e BMW, vêm desenvolvendo projetos como o sistema de controle veicular baseado em voz (CHANG et al., 2009) e controle de *infotainment* multimodal (ALTHOFF et al., 2005) que utilizam de fala ou de gestos podem facilitar a interação do passageiro com o veículo.

2.3 AGL - AUTOMOTIVE GRADE LINUX

O Automotive Grade Linux é um projeto de código aberto (ou open-source), lançado em 2012 pela fundação Linux Foundation, que visa o desenvolvimento de um sistema operacional padrão, baseado no kernel Linux, para produtos de *In-Vehicle Infotainment* (IVI) (AGL, 2012) e mais recentemente *Internet of Vehicles* (KLAVMARK; VIKINGSSON, 2015).

O AGL define uma plataforma base para o desenvolvimento de aplicações, mostrado na Figura 2. Este modelo define as camadas oferecidas para acesso ao hardware, diferentes protocolos de comunicação e interfaces humano-máquina. Esta arquitetura foi baseada no sistema *Tizen IV* da Samsung® e, portanto, oferece funcionalidades já prontas de interface gráfica, que podem ser estendidas ou modificadas utilizando apenas Javascript ou HTML5 (KLAVMARK; VIKINGSSON, 2015).

Esta plataforma pode comunicar-se com o veículo através de SocketCAN ou uma interface de alto-nível chamada de AMB (do inglês *Automotive Message Broker*),

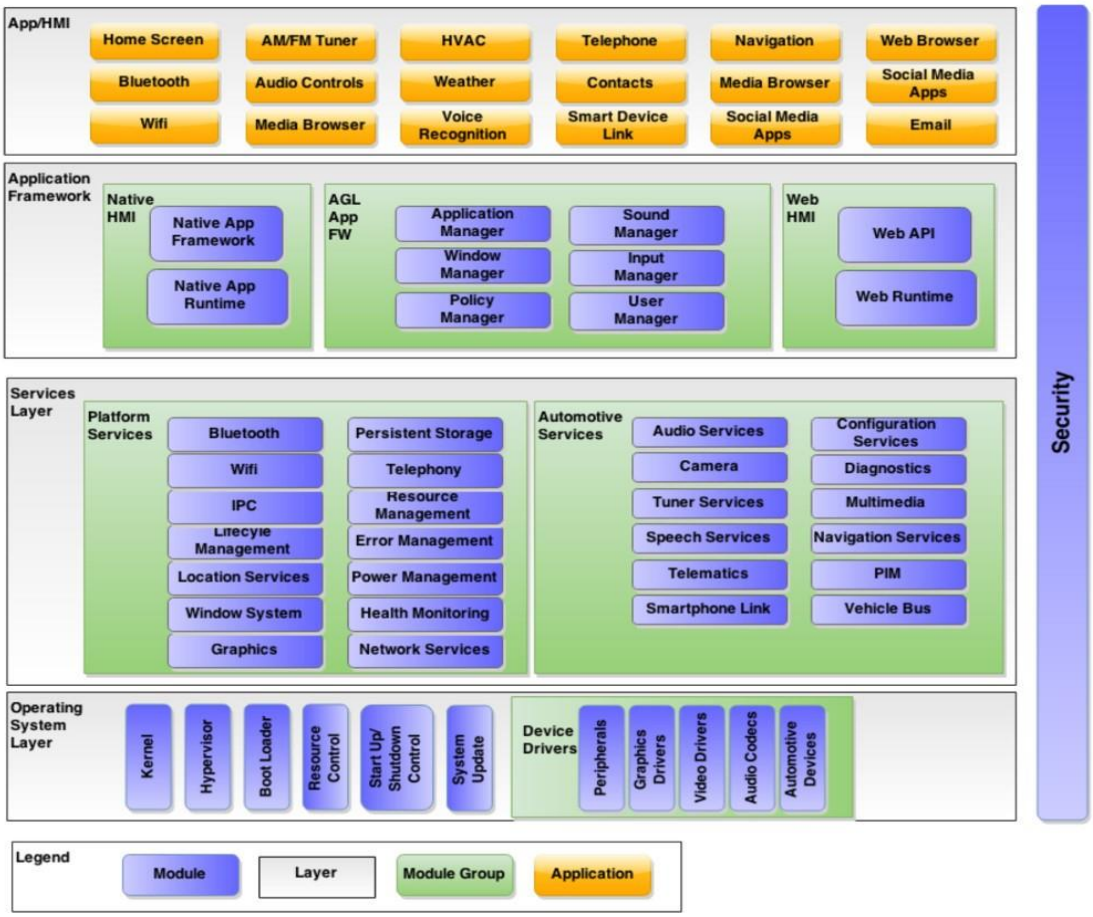

Figura 2 - Arquitetura de software do AGL.

Fonte: AGL (2012)

que oferece uma interface padrão de acesso a qualquer dispositivo interno do veículo.

Desde o início de seu projeto várias marcas da indústria automotiva, como Jaguar®, Land Rover®, Nissan® e Toyota® juntaram-se à iniciativa a fim de acelerar o desenvolvimento e a aplicação para IVI. Em 2018 é esperado que a Toyota® coloque no mercado o primeiro veículo com AGL para IVI (TOYOTA, 2017), iniciando uma nova etapa no desenvolvimento do sistema-operacional.

2.4 YOCTO PROJECT

O projeto Yocto (ou em inglês Yocto Project) é um projeto de código aberto criado e mantido pela fundação Linux Foundation (FOUNDATION,), que tem como

objetivo produzir ferramentas e processos que facilitem a criação de sistemas embarcados baseados em Linux para diversas arquiteturas, como ARM®, MIPS®, PowerPC® e x86/64.

Uma das partes chave deste projeto é a integração do sistema de construção de imagens OpenEmbedded, que oferece ferramentas para facilitar a customização, construção e manutenção de uma distribuição Linux própria específica para cada projeto embarcado.

Além da criação de imagens, as ferramentas do Yocto permitem a criação de ferrametas de desenvolvimento (SDKs), além da possibilidade de adicionar testes de regressão e sanidade ao fim da compilação da imagem ou até mesmo a criação de máquinas virtuais QEMU para testes de unidade.

3 DESENVOLVIMENTO DO TEMA

Neste capítulo serão apresentados os detalhes de implementação de um sistema IoV baseado em AGL para uma aplicação IVI seguindo a UCB 2.0, descrevendo como foi obtido uma base de desenvolvimento de aplicações.

3.1 HARDWARE

Para o desenvolvimento da aplicação de testes foi necessária a preparação de um hardware específico que fosse compatível com a versão do AGL 4.0.1 (Daring Dab) para realizar o papel de CMCU do ambiente IoV. Dentre os hardwares acessíveis, a *Raspberry Pi* 3 Modelo B foi escolhida. Este modelo da placa de desenvolvimento é composto pelo seguinte hardware:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1 GB RAM
- 4 portas USB 2.0
- 40 pinos GPIO • Porta HDMI
- Porta Ethernet
- CSI/DSI
- Slot micro-SD
- Núcleo de vídeo VideoCore IV 3D

Entretanto, este modelo, por ser de uso educacional, não conta com interfaces específicas para IoV, como CAN ou LIN para comunicação intra-veiculares

e 3G/4G para comunicação com servidores remotos ou outros veículos. Por isso, foi necessária a inclusão de outros hardwares: módulo de interface CAN e módulo de interface 4G.

A interface CAN foi obtida através do módulo de interface PiCAN 2, da empresa Cooperhill, com ele é possível utilizar o barramento CAN 2.0 do veículo através da interface OBD 2.0 através do módulo CAN Bus MCP 2515, com ele é possível a criação de aplicações em C/C++ ou Python (através de SocketCAN ou do AMB).

Para o módulo de interface 4G foi optado pela utilização de um aparelho celular smartphone *Samsung Galaxy® S7 Edge* em modo “Roteador Wi-Fi”, configurado como mostrado na Figura 4. Desta forma, utilizando-se de umas das portas USB 2.0 da Raspberry Pi 3, foi possível conexão com a internet utilizando o 4G do aparelho. A Figura 3 mostra o conjunto completo de desenvolvimento

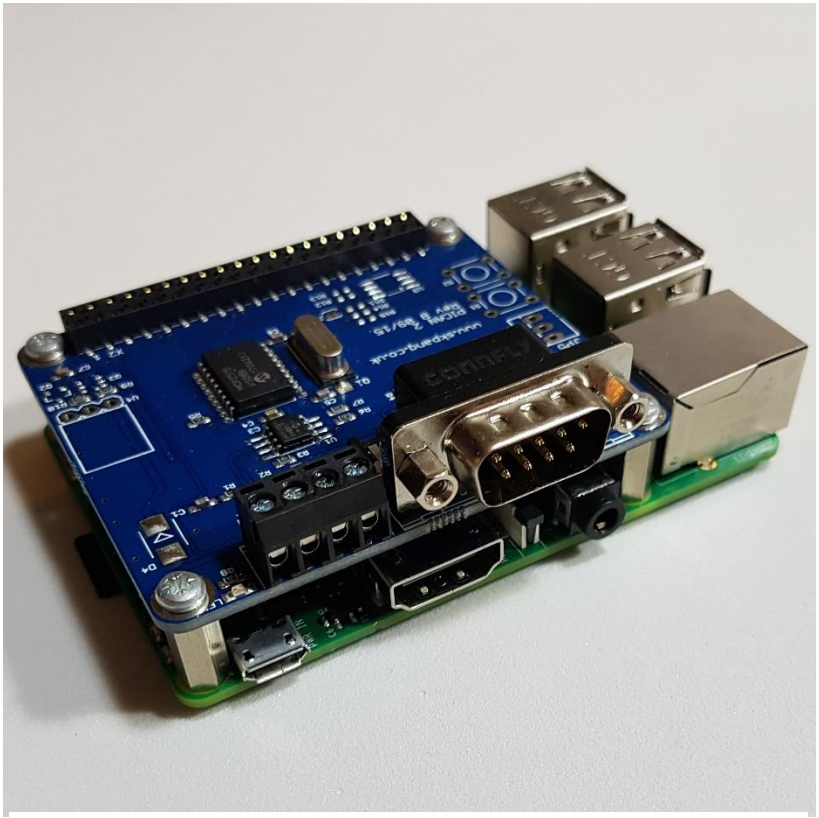

Figura 3 - Conjunto completo do módulo de desenvolvimento

Fonte: Autoria própria

utilizado nos testes.

Para o desenvolvimento do firmware e das aplicações foi feita a utilização de um computador pessoal utilizando Ubuntu 16.04.3. Na próxima sessão será descrito

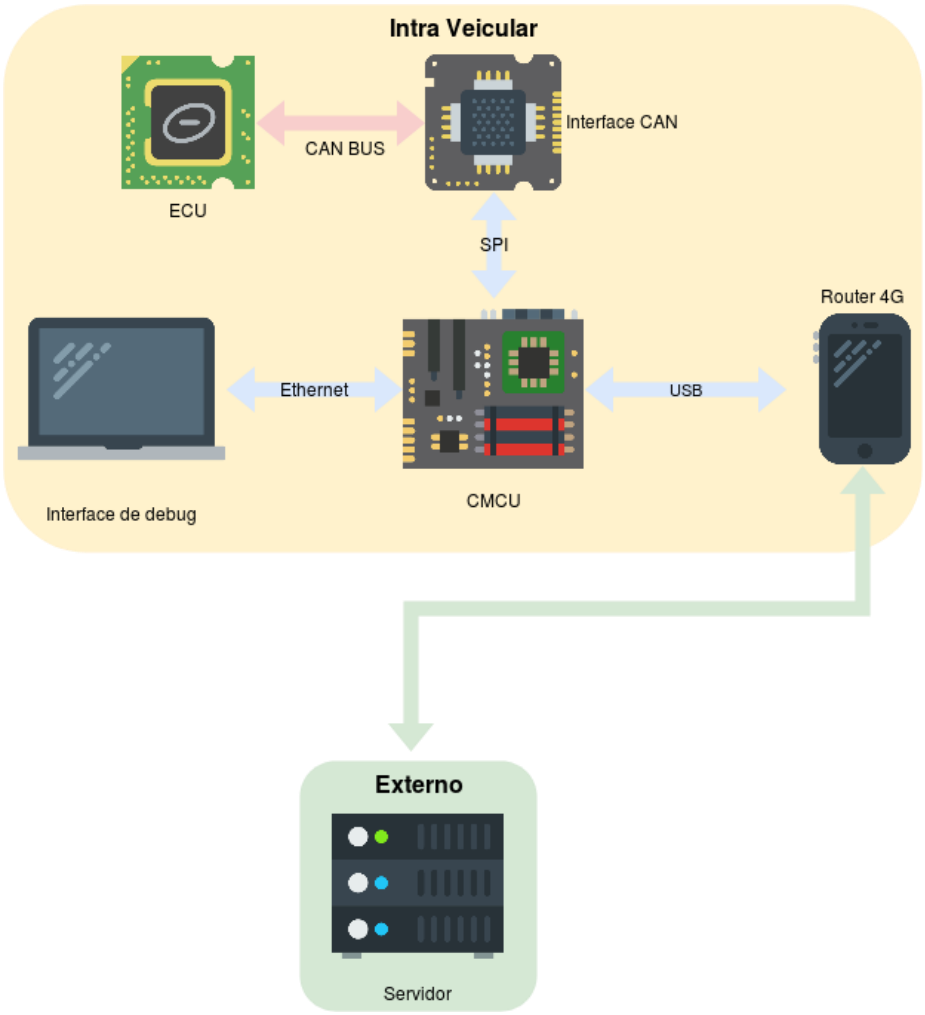

Figura 4 - Esquema de ligação da CMCU

Fonte: Autoria própria

como este hardware foi mapeado em um firmware gerado através das ferramentas Yocto, utilizando as ferramentas AGL 4.0.1, seguindo o UCB 2.0.

3.2 FIRMWARE

Para a geração do firmware deve ser feita a utilização da ferramenta de geração de imagens Linux embarcado Yocto Project 2.2 (Morty). Como o AGL não é padrão dentro dos repositórios da distribuição, é necessária a inclusão do código-fonte deste junto ao repositório.

Por padrão o AGL já disponibiliza o driver de interface CAN Bus MCP 2515, configurando o oscilador 1,6 MHz e o pino de interrupção 25 do GPIO. As ferrametas

SocketCAN, AMB e também o can-utils são disponibilizados por padrão pelo AGL, sem necessidade de integrá-los ao ambiente Yocto.

A interface PythonCAN para SocketCAN deve ser adicionada ao firmware para o desenvolvimento de aplicações em Python. A próxima sessão descreverá o processo de criação da aplicação de testes e do framework base para desenvolvimento de aplicações IoV em AGL, utilizando os recursos adicionados à imagem pelo Yocto.

3.3 FRAMEWORK

A Figura 5 mostra a arquitetura de software definida para uma aplicação IoV utilizando AGL. Os quadros em azul compreendem a implementação do framework. “IoV App” é a aplicação principal, que trata a lógica de captura, filtragem, pré-processamento para transmissão ao servidor.

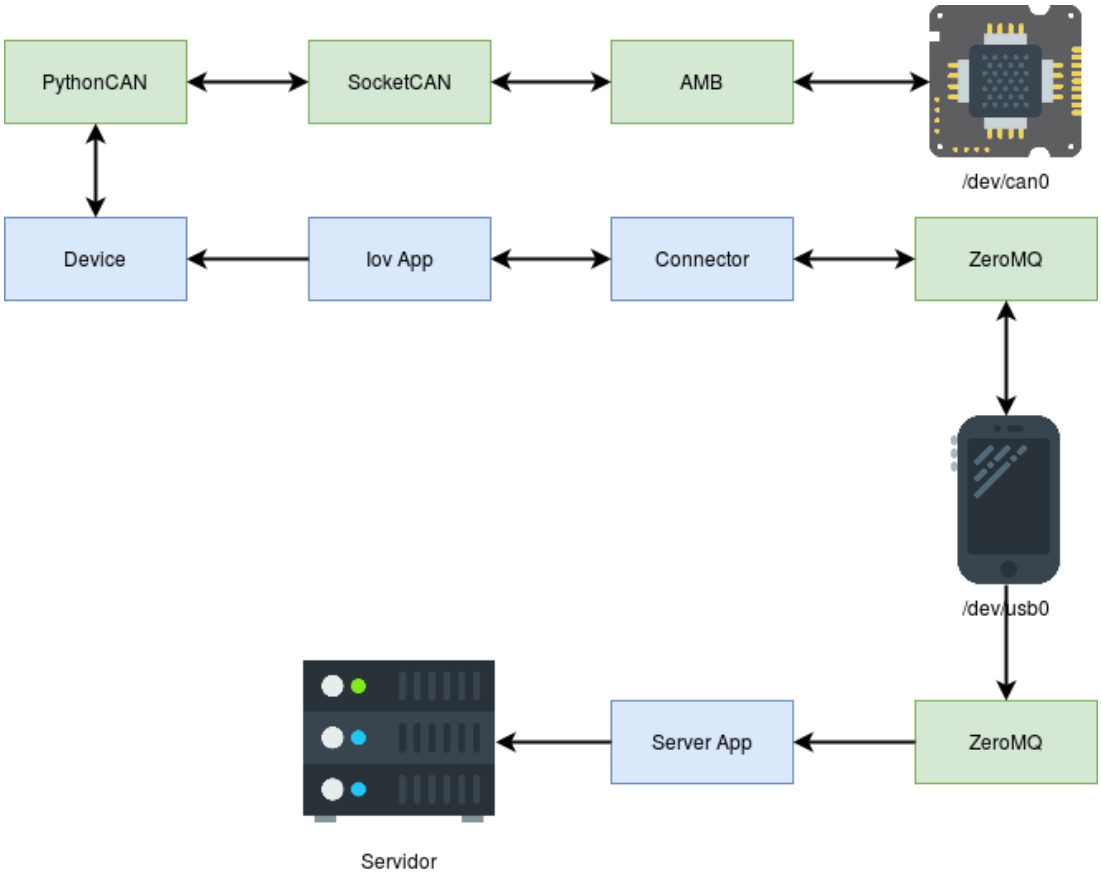

Figura 5 - Arquitetura de software para o framework proposto

Fonte: Autoria própria

O bloco “Connector” trata de forma genérica a comunicação entre a CMCU e o servidor de dados ou processamento, abstraindo a interface de transmissão em métodos comuns. Neste caso, o bloco de Connector está ligado a biblioteca ZeroMQ, onde é feita a transmissão dos dados via conexão TCP/IP abstraída em filas de entrada e saída.

Do lado interno do veículo, o bloco “Device” é responsável por abstrair dispositivos de dados internos do veículo, ou seja, sensores e atuadores conectados à ECU via CAN ou LIN. Para este trabalho considera-se apenas a utilização da interface CAN, via PythonCAN. Esta interface é capaz de escrever e ler no barramento CAN via SocketCAN, aqui abstraído pelo AMB da AGL.

O lado do servidor contém a segunda parte do conjunto ZeroMQ para troca de dados remota. O bloco “Server App” abstrai toda a gravação e processamento dos dados. Neste momento, o servidor tem apenas vias de entrada, ou seja, apenas a CMCU pode se comunicar com o servidor, porém o servidor não é capaz de realizar a comunicação com CMCU.

3.4 APLICAÇAO DE TESTES

Utilizando a imagem e o framework proposto, uma aplicação de testes foi definida para demonstrar o funcionamento do conjunto hardware e software para IoV. Esta aplicação realiza a captura de informações de um veículo automotor Peugeot® 308 Allure 2012/2013 1.6 flex e envia os dados para serem armazenados e processados num servidor. O processo de captura realiza a leitura dos dados à cada 5 segundos, com o carro em movimento. Quando um conjunto de leitura é finalizado, ou seja, dados de todos os sensores forem colhidos, uma mensagem é enviada ao servidor.

Este servidor roda uma aplicação que aguarda o envio de dados da CMCU e quando estes dados são enviados, o servidor salva em um arquivo local até atingir um limite mínimo de 90 observações. Os dados armazenados são:

- Rotação do motor
- Temperatura do motor
- Velocidade do veículo

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste capítulo serão apresentados os resultados obtidos através da montagem e testes do hardware, firmware e da aplicação de testes seguindo o framework proposto no capítulo anterior. Na primeira sessão será feita a descrição da aplicação obtida através da arquitetura proposta anteriormente. A última sessão irá mostrar os detalhes dos dados obtidos através dos testes em forma de gráficos.

4.1 APLICAÇÃO DO FRAMEWORK

O framework foi aplicado através do desenvolvimento da aplicação de testes proposta anteriormente. Com isto, foi obtida uma base de código inicial para projetos futuros. O código foi disponibilizado como código aberto no link em Anexo 1. O diagrama mostrado na Figura 6 mostra os detalhes da implementação e a relação

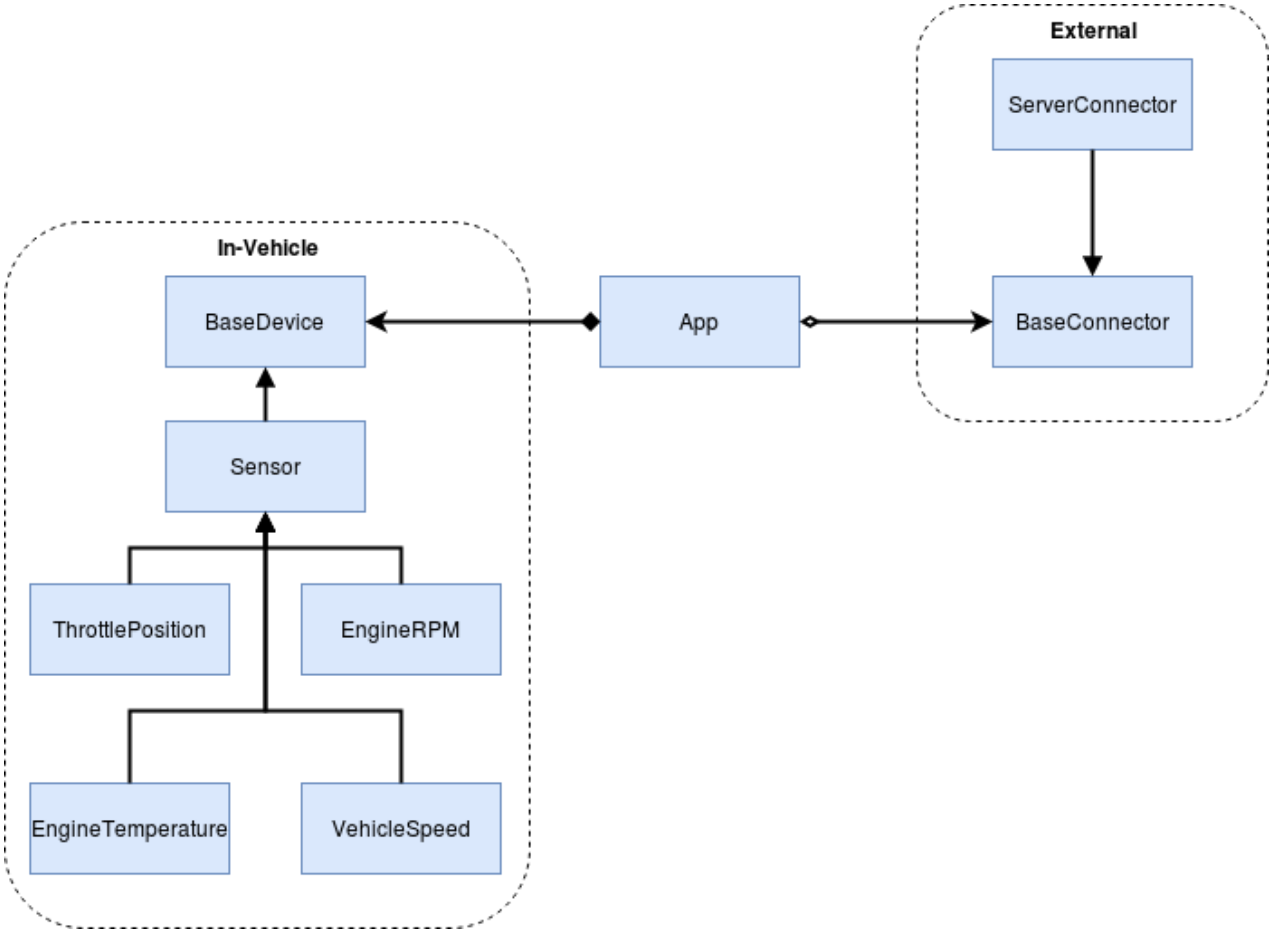

Figura 6 - Arquitetura de software para o framework proposto

Fonte:Autoria própria

entre as classes.

Como é possível observar, foram implementadas classes de abstração para os dispositivos internos do veículo, chamada de “BaseDevice”, esta classe foi estendida para uma abstração mais específica de sensores, chamada de “BaseSensor”. Desta surgiram 3 classes específicas para diferentes sensores: “EngineRPM” para obtenção de valores de rotação do motor, “EngineTemperature” para obtenção da temperatura do motor e “VehicleSpeed” para a velocidade.

A classe de “App” define a lógica de repetição de captura e envio de dados para o servidor. Este envio é feito através da utilização da classe “ServerConnector”, que implementa uma abstração para conexão via ZeroMQ ao servidor de dados. Esta classe estende a classe “BaseConnector”, que oferece interface comum para envio de dados entre diferentes protocolos e dispositivos.

4.2 TESTES

Os dados armazenados no servidor através da comunicação com o CMCU foram obtidos através de testes de viagem dentro da cidade de Curitiba no caminho especificado pelo mapa mostrado na Figura 7, sem paradas ou mudanças de trajetos.

Figura 7 - Mapa da rota de testes

Fonte: Autoria própria

O caminho definido tinha distância total de trajeto de 3 km, contendo 8 semáforos (representados com o ícone vermelho no mapa), 7 cruzamentos de parada obrigatória (representados pelos ícones em amarelo), 1 radar de 40 km/h (representado em verde). O trajeto foi executado em cerca de 10 minutos por bateria, sendo executadas 3 baterias de testes entre 7 e 9 de setembro de 2017. Os gráficos abaixo mostram os dados obtidos em forma de gráficos para melhor visualização, sendo a Figura 8 gráficos do primeiro dia de teste, Figura 9 do segundo dia e Figura 10 do último dia.

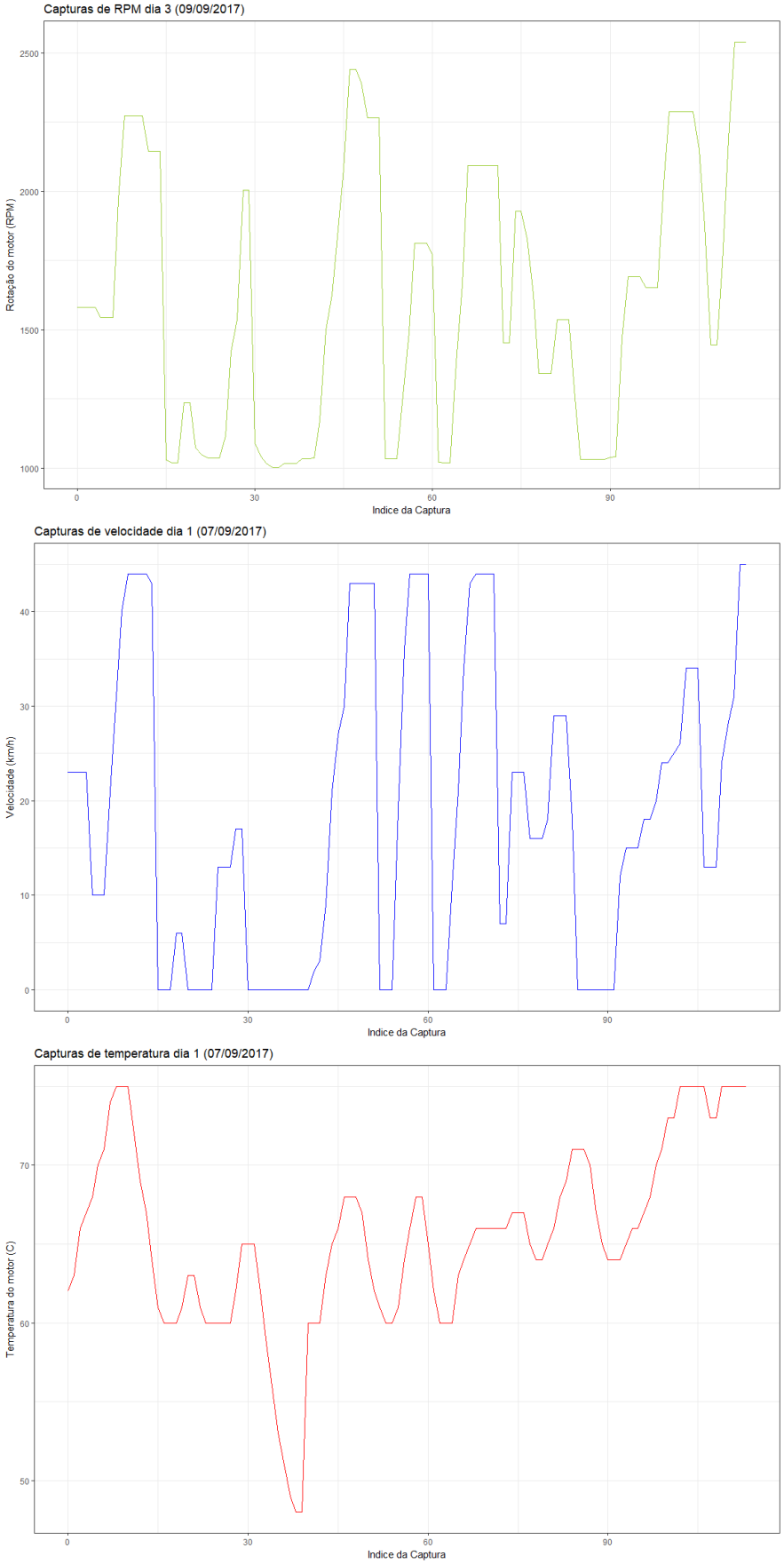

Figura 8 - Gráfico de capturas relativas ao primeiro dia de testes (velocidade, RPM e temperatura)

Fonte: Autoria própria

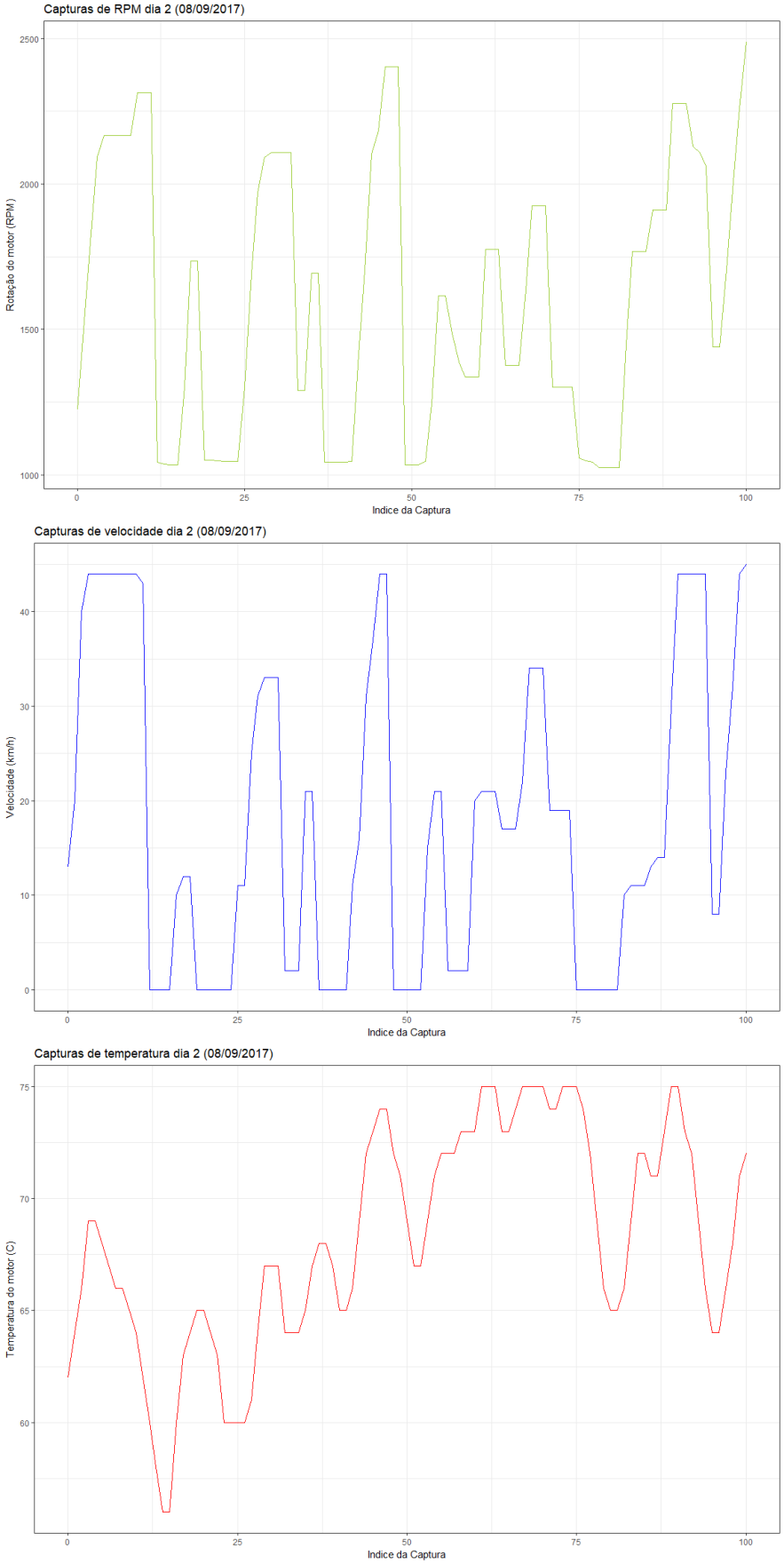

Figura 9 - Gráfico de capturas relativas ao segundo dia de testes (velocidade, RPM e temperatura)

Fonte: Autoria própria

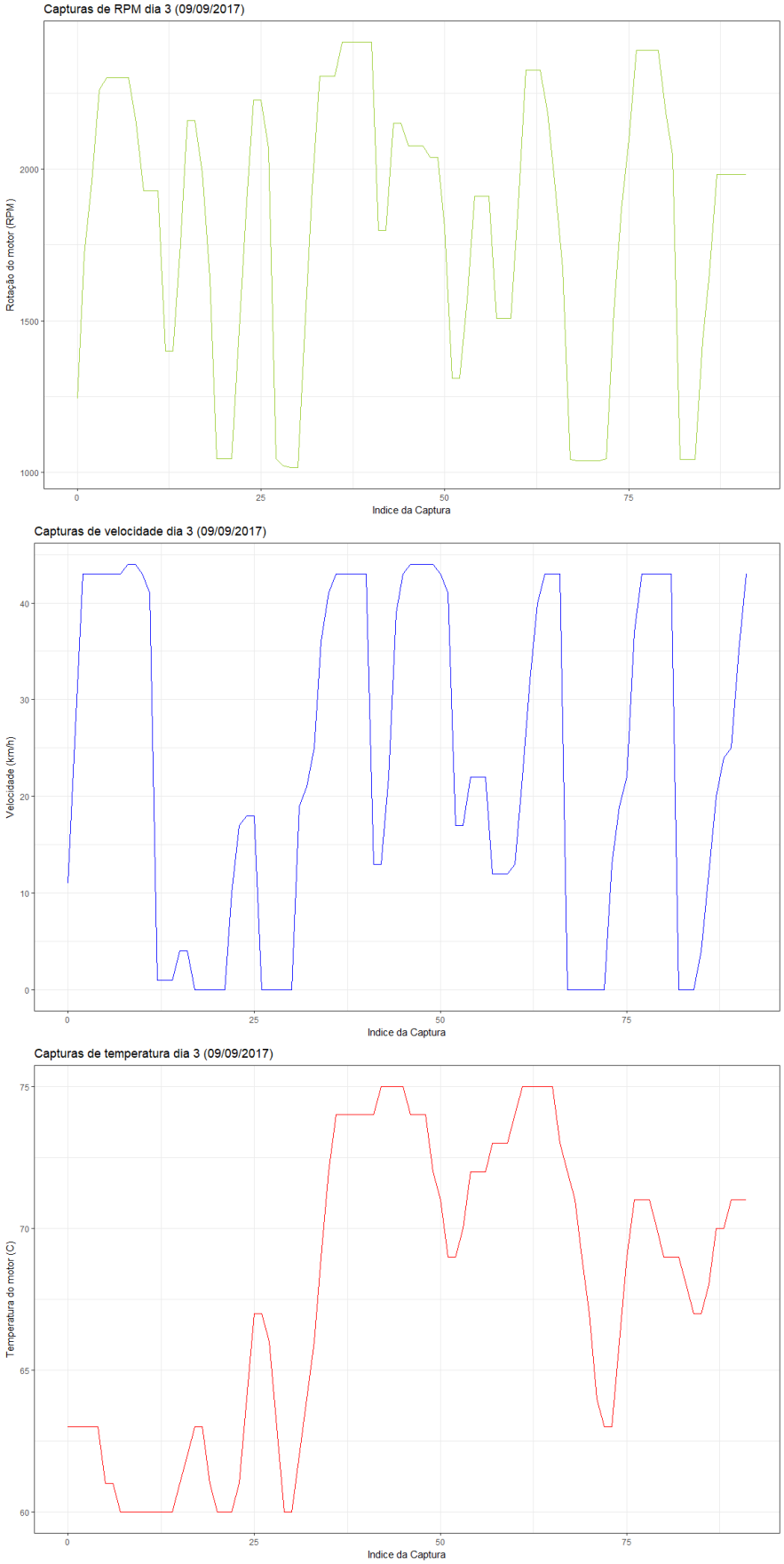

Figura 10 - Gráfico de capturas relativas ao terceiro dia de testes (velocidade, RPM e temperatura)

Fonte: Autoria própria

E possível perceber o comportamento correto da velocidade do veículo correlacionado à rotação e a temperatura do motor. Estes gráficos foram gerados utilizando o programa R, através da exportação dos dados gravados no servidor em um arquivo CSV. E possível observar´ a diferença no tamanho das capturas entres os 3 dias de testes, isso se deve à problemas de comunicação com servidor e diferenças no tempo de trajeto devido a variáveis de tráfego (tempo de espera no semáforo, veículos na pista, tempo de reação, etc.). Tabela 1 mostra a quantidade de dados perdidas nos 3 dias de testes devido à problemas de comunicação com servidor de dados.

Tabela 1 - Resultado dos testes

REFERÊNCIA	DIA	CAPTURAS	ERROS	TAXA DE ERRO (%)
1	07/09/2017	114	7	6,14
2	08/09/2017	101	6	5,61
3	09/09/2017	92	5	5,15

É importante ressaltar também que podem haver discrepâncias entre os valores de captura e os valores reais, que podem ocorrer devido à conversão dos dados obtidos da rede CAN, problemas de transmissão com o servidor ou até mesmo problema na gravação no banco de dados. Apesar disto, todos os dados obtidos podem ser considerados íntegros e consistentes com o trajeto esperado, mostrando que mesmo sem o tratamento adequado não houveram graves problemas de comunicação ou armazenamento.

5 CONSIDERAÇÕES FINAIS

Aplicações de Internet das coisas focada no uso veicular, chamada de *Internet of Vehicles* (IoV), tem sido um importante assunto dentro do desenvolvimento de novas aplicações e tecnologias dentro da indústria automotiva. Dentre estas, destaca-se o esforço realizado pela Linux Foundation na criação do *Automotive Grade Linux* (AGL), que vem sendo adotado por boa parte da indústria como uma plataforma base para desenvolvimento de aplicações de informação e entretenimento automotivo.

Este trabalho realizou a criação de um framework básico para utilização do AGL em um hardware popular, chamado de CMCU, contando com uma Raspberry Pi 3 e com um módulo de comunicação CAN, para assim facilitar o desenvolvimento de novos produtos e pesquisas baseados em AGL. O framework foi implementado sob a solução Yocto e utilizando apenas softwares de código aberto, não atrelando o uso a implicações legais.

A aplicação desenvolvida para testes mostra a capacidade do sistema desenvolvido pela Linux Foundation e a facilidade da criação de aplicações que possam utilizar dados de um veículo para qualquer tipo de processamento na nuvem, utilizando apenas um celular como central de comunicação. O desenvolvimento da aplicação mostrou a capacidade do AGL em entregar ferramentas base para comunicação (como a libsocketcan) e a capacidade de se utilizar linguagens de alto-nível (como Python), o que pode acelerar o processo de prototipação de uma aplicação de IoV. Além disto, a vasta gama de protocolos já suportados pelo kernel Linux facilita a adaptação de novos sensores e atuadores veiculares.

Apesar disto, tanto o AGL como a IoV são conceitos novos que precisam ser refinados e melhorados com dados obtidos de campo para que tenhamos soluções simples, modulares e baratas para a indústria. Trabalhos futuros podem utilizar este conceito e as ferramentas desenvolvidas aqui para implementar novas soluções e colaborar com o desenvolvimento da plataforma.

REFERÊNCIA(S)

AGL. **Announcing Automotive Grade Linux**. 2012. Disponível em: <https://goo.gl/3Nvb4D>. Acessado em: 8 de junho de 2017.

TOYOTA. **Automotive Grade Linux Platform Debuts on the 2018 Toyota Camry.** 2017. Disponível em : <https://goo.gl/NjTjpn >. Acessado em: 8 de junho de 2017.

ALTHOFF, F. et al. **Robust multimodal hand-and head gesture recognition for controlling automotive infotainment systems**. VDI BERICHTE, VDI; 1999, v. 1919, p. 187, 2005.

BELYAEV, E. et al. **Live video streaming in ieee 802.11 p vehicular networks: demonstration of an automotive surveillance application**. In: IEEE. Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. [S.l.], 2014.

CHANG, J. C. et al. **Usability evaluation of a volkswagen group in-vehicle speech system**. In: ACM. Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications. [S.l.], 2009. p. 137–144.

COUNTS, S. **Automotive Grade Linux will be the backbone of your connected car**. 2017. Disponível em : <http://www.autoblog.com/2017/01/06/automotive-grade-linux-connected-car-backbone/>. Acessado em: 8 de junho de 2017.

FANGCHUN, Y. et al. **An overview of internet of vehicles**. China Communications, IEEE, v. 11, n. 10, p. 1–15, 2014.

FOUNDATION, L. **The Linux Foundation Announces Yocto Project Steering Group and Release 1.0 - The Linux Foundation**. Disponível em: <https://goo.gl/otNyH7>. Acessado em: 10 de março de 2017.

GAI, P.; VIOLANTE, M. **Automotive embedded software architecture in the multi-core age**. In: IEEE. Test Symposium (ETS), 2016 21th IEEE European. [S.l.], 2016.

GERLA, M. et al. **Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds**. In: IEEE. 2014 IEEE World Forum on Internet of Things (WF-IoT). [S.l.], 2014.

KERNER, S. M. **Linux Foundation Accelerates Automotive Grade Linux**. 2016. Disponível em: <http://www.eweek.com/mobile/linux-foundation-accelerates-automotive-grade-linux>. Acessado em: 8 de junho de 2017.

KLAVMARK, A.; VIKINGSSON, T. **Study on open source in-vehicle infotainment (ivi) software platforms**. 2015.

LOWREY, L. H. et al. **Internet-based vehicle-diagnostic system**. [S.l.]: Google Patents, 2003. US Patent 6,611,740.

NEWCOMB, D. **The next big os war is in your dashboard**. Revista Wired, Abril de 2012.

SUNDMAEKER, H. et al. **Vision and challenges for realising the internet of things**. Cluster of European Research Projects on the Internet of Things, European Commision, 2010.

WAN, J. et al. **Mobile crowd sensing for traffic prediction in internet of vehicles**. Sensors, Multidisciplinary Digital Publishing Institute, v. 16, n. 1, p. 88, 2016.