



## Console Touchscreen para Indicadores de Destino

**CALLEBE SOARES BARBOSA**

Julho de 2018

CALLEBE SOARES BARBOSA

**CONSOLE TOUCHSCREEN PARA INDICADORES DE  
DESTINO**



PORTO

2018

# CONSOLE TOUCHSCREEN PARA INDICADORES DE DESTINO

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de DSEE - Dissertação do Mestrado em Engenharia Eletrotécnica - Sistemas Elétricos de Energia.

**Candidato:** Callebe Soares Barbosa  
Nº 1170132

**Orientador:** João Miguel Queirós Magno  
Leitão

**Empresa:** DMS - Display & Mobility Solutions

PORTO

2018

Dedico ao Deus Altíssimo, o qual concede-me diariamente a dádiva da vida, para através dela lhe render louvor.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por prover a mim os meios pelos quais fui, e sou, sustentado todos os dias. Por me dar a oportunidade e a capacitação para realização deste trabalho. Em seguida agradeço a estrutura que formou meu carácter e me deu suporte em todas as minhas empreitas; a minha querida família. Ao meu pai José Geral, a minha mãe Maria Rosa Barbosa e ao meu irmão Victor E. Soares Barbosa. Sem esta família sei que jamais teria chegado até aqui.

Agradeço também ao meu professor orientador Dr. João Miguel Queirós Magno Leitão, pela paciência, dedicação e respeito com que me direcionou e ensinou ao longo da realização deste trabalho. Agradeço também a banca pelos conselhos, críticas e correções, que contribuem em muito no meu aperfeiçoamento profissional.

Agradeço aos meus colegas da DMS (*Displays & Mobility Solutions*), os quais ao longo de meu estágio acadêmico me receberam com paciência, compreensão e companheirismo. Em especial ao meu supervisor de estágio o Eng. José Semanas, o qual me propôs este tema de dissertação, e me apresentou ao departamento de desenvolvimento da DMS. Aos colegas Msc. Márcio Marques Correia, ao Eng. Carlos Silva e aos Projetistas Bruno Maia e Bruno Monteiro, que me auxiliaram diretamente no desenvolvimento deste trabalho, fica aqui registrado minha eterna gratidão.

Deixo também aqui registrado minha gratidão e reconhecimento a toda a Universidade Tecnológica Federal do Paraná (UTFPR). A esta instituição que me recebeu como aluno, e me deu a oportunidade de estudar junto a um grupo docente de excelência, e que me preparou para o nobre exercício da engenharia.

Por fim, agradeço aos meus queridos amigos, que surgiram ao longo de toda a minha jornada e que de maneira extraordinária comburiram para meu crescimento, celebrando com minhas vitórias e me sustentando em minhas dificuldades. Em especial aos meu queridos amigos da UTFPR; o Eng. Erivelto Sauzen Müller, Eng. Leandro Fabian e o Eng. Matheus Toccolini.

## RESUMO

SOARES BARBOSA, Callebe. Console Touchscreen Para Indicadores de Destino. 2018. 69p. Dissertação de Mestrado - Curso de Mestrado em Engenharia Eletrotécnica, Instituto Superior de Engenharia do Porto. Porto, 2018.

Os Indicadores de Destino são Sistemas de Informação ao Público mais comuns no dia-a-dia, sendo utilizados principalmente em autocarros, metros e trens. Um sistema de indicadores de destino é tipicamente constituído de um console e um conjunto de painéis eletrônicos onde as informações ao público são exibidas. Com o avanço de novas tecnologias na áreas de sistemas embarcados, novas soluções baratas tem surgido para tornar estes sistemas mais inteligentes e eficiente. Dentre elas esta a utilização de *displays touchscreen* em interfaces mais amigáveis e práticas ao usuário. O console é o dispositivo responsável pela interface do usuário, e também pelo controle dos painéis eletrônicos do sistema. Neste trabalho é desenvolvido, com base em um modelo atual, um novo console dotado de uma interface *touch screen* intuitiva, editor de informações visuais dos painéis, conectividade com Wifi e um sistema embarcado Linux, que possibilita que novas funcionalidade sejam adicionadas futuramente.

**Palavras-chave:** Console, Touchscreen, Raspberry Pi, Sistema de Informação ao Público.

## ABSTRACT

SOARES BARBOSA, Callebe. Touch Screen Console for Destination Indicators. 2018. 69p. Master's Dissertation - Master's Degree in Electrotechnical Engineering, Instituto Superior de Engenharia do Porto. Porto, 2018.

Destination Indicators are the most common Public Information Systems, being mainly used in buses, subways and trains. An Destination Indicators system usually consists of a console and a set of electronic panels where information for the public is displayed. With the advancement of new technologies in the embedded systems areas, new cheap solutions have create to make these systems more smart and efficient. The use of touch screen displays in a new friendly and practical interface is one of them. The console is the device responsible for the user interface and control of the system's electronic panels. Based on a current model, a new console is developed with an intuitive touch screen interface, panel information editor, Wifi connectivity and a Linux kernel, that allows new functionality to be added in future.

**Keywords:** Console, Touchscreen, Raspberry Pi, Public Information Systems.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	ESTRUTURA DO TRABALHO	2
1.2	RECURSO UTILIZADOS	3
1.3	OBJETIVOS	3
1.3.1	Objetivos Gerais	3
1.3.2	Objetivos Específicos	4
<b>2</b>	<b>TECNOLOGIAS ENVOLVIDAS</b>	<b>7</b>
2.1	DISPLAY <i>TOUCHSCREEN</i> TOPWAY - HKT080ATA-C	7
2.2	RASPBERRY PI ZERO W	9
2.3	INDICADORES DE DESTINO - DMS	12
2.4	INTERFACE DE COMUNICAÇÃO UART-RS485	14
2.5	GESTOR DE INDICAÇÃO DE DESTINOS	16
2.5.1	Protocolo LDESIGN	20
<b>3</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO DMSCONSOLE</b>	<b>23</b>
3.1	GERENCIADOR XML	23
3.2	GERENCIADOR DE INTERFACE	26
3.2.1	Página de Seleção de Destino	28
3.2.2	Página de Seleção de Linha	29
3.2.3	Página do Menu de Configurações	30
3.2.4	Página de Seleção de Fontes	31
3.2.5	Página de Edição de Fontes	32
3.2.6	Página de Configuração dos Painéis	34
3.2.7	Página de Configuração do <i>Display</i>	35
3.2.8	Gerenciador Painéis	36
3.2.8.1	MSG Network Config	36
3.2.8.2	MSG Define Brightness	37
3.2.8.3	MSG Reset	37

3.2.8.4	MSG INFO .....	38
3.2.8.5	MSG Panel Test .....	42
3.2.8.6	ACK e NACK .....	43
3.2.8.7	Panel ID .....	43
3.2.9	Bit Bang para Comunicação UART-RS485 .....	43
<b>4</b>	<b>MONTAGEM E TESTE DO PROTÓTIPO .....</b>	<b>47</b>
4.1	DESENVOLVIMENTO DA PLACA DE INTEGRAÇÃO .....	47
4.2	REALIZAÇÃO DOS TESTES .....	54
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>59</b>

## LISTA DE FIGURAS

Figura 1:	Diagrama do Sistema de Indicadores de Destino para o Novo Console - DMS . . . . .	4
Figura 2:	<i>Display Touchscreen</i> TOPWAY HKT080ATA-C . . . . .	7
Figura 3:	Programa de Edição de Interface <i>TOPWAY TML Graphics Editor</i> . . . . .	8
Figura 4:	Raspberry Pi Zero W . . . . .	10
Figura 5:	Indicador de Destino <i>SICON II</i> - DMS . . . . .	12
Figura 6:	Autocarro Modelo - 2 Painéis Eletrônicos de Indicação de Destino . . . . .	13
Figura 7:	Diagrama do Sistema de Indicadores de Destino - <i>DMS</i> . . . . .	13
Figura 8:	Console do Indicador de Destino <i>SICON II</i> - DMS . . . . .	14
Figura 9:	Interface de Comunicação UART RS-485 . . . . .	15
Figura 10:	Comprimento da Rede Vs Taxa de Transferência . . . . .	16
Figura 11:	Tipos de Topologia para UART-RS485 . . . . .	16
Figura 12:	Gestor de Indicadores de Destino - Tela Inicial . . . . .	17
Figura 13:	Linhas de Autocarro da STCP . . . . .	18
Figura 14:	Estrutura de Dados do GID . . . . .	19
Figura 15:	Gestor de Indicadores de Destino . . . . .	20
Figura 16:	Interface no <i>Display Touchscreen</i> DMS . . . . .	27
Figura 17:	Fonte de Dados - <i>SICON II</i> . . . . .	28
Figura 18:	Fonte de Dados - Novo Console . . . . .	28
Figura 19:	Página de Seleção de Destino - <i>DMSConsole</i> . . . . .	29
Figura 20:	Página de Seleção de Linha - <i>DMSConsole</i> . . . . .	30
Figura 21:	Página do Menu de Configurações - <i>DMSConsole</i> . . . . .	31
Figura 22:	Página de Seleção de Fonte - <i>DMSConsole</i> . . . . .	32
Figura 23:	Editor de Fontes <i>DMSConsole</i> . . . . .	33
Figura 24:	Página de Configuração dos Painéis - <i>DMSConsole</i> . . . . .	34

Figura 25:	Página de Configuração do Console - DMSConsole . . . . .	35
Figura 26:	MSG Info - Page General Info . . . . .	39
Figura 27:	MSG Info - Page Info Header . . . . .	40
Figura 28:	Conversor UART TTL - UART 485 . . . . .	45
Figura 29:	Esquemático do Conversor CC-CC . . . . .	48
Figura 30:	Esquemático do Conversor UART TTL / UART RS232 . . . . .	49
Figura 31:	Esquemático do Conversor UART TTL / UART RS485 . . . . .	49
Figura 32:	Esquemático Raspbery Pi Zero W . . . . .	50
Figura 33:	Layout de Montagem DP0507-0A0 - Superior . . . . .	50
Figura 34:	Layout de Montagem DP0507-0A0 - Inferior . . . . .	51
Figura 35:	Modelo 3D DP0507-0A0 - Inferior . . . . .	51
Figura 36:	Modelo 3D DP0507-0A0 - Inferior . . . . .	51
Figura 37:	Modelo 3D Console Sicon III . . . . .	52
Figura 38:	Protótipo SICON III - Lateral Esquerda . . . . .	53
Figura 39:	Protótipo SICON III - Lateral Direita . . . . .	53
Figura 40:	Painel Eletrônico da Rede de Indicadores de Destino - Sem Efeito de Rolagem . . . . .	55
Figura 41:	Painel Eletrônico da Rede de Indicadores de Destino - Com Efeito Rolagem . . . . .	56

## LISTA DE TABELAS

Tabela 1:	Configuração GPIO - Raspberry Pi Zero . . . . .	11
Tabela 2:	Formato Geral - Protocolo Geral de Comunicação LDESIGN . .	21
Tabela 3:	Formato Geral - MSG Info . . . . .	38
Tabela 4:	Formato Page Info - MSG Info . . . . .	40
Tabela 5:	Formato Panel ID . . . . .	43

## 1 INTRODUÇÃO

Segundo Cunha e Lacomblez (2006), o sistema de transporte público português, por meio de autocarros, apresenta desde 2006 um elevada taxa de expansão e um crescimento no volume de investimentos em melhorias na qualidade do serviço prestado, visando um aumento no número de clientes e na fidelização no transporte público. Dentre as medidas adotadas estão o aumento no número unidades de autocarros, o aumento no número de linhas oferecidas, implementação do sistemas de assinatura e melhorias nos sistemas de informação ao público.

Os sistemas de informação pública, ou Public Information System (PIS), segundo Orman (1989), são sistemas desenvolvidos para o uso do público em geral, com o objetivo transmitir informações de utilidade pública. Em comparação aos sistemas de informação utilizadas no meio comercial ou particular, como em fábricas ou empresas, os PISs possuem um maior impacto social, econômico e estrutural político, através da mudança de acessibilidade e disponibilidade de informações críticas ao público em geral(ORMAN, 1989).

No contexto do sistema de transporte público os PISs são elementos comumente presentes. Nas estações de comboios da CP (2018), em Portugal, os horários de partida e chegadas de cada comboio é informado através de um PIS que apresenta estas informações nos conjuntos de painéis eletrônicos espalhados pela estação. Já nos pontos de paragem de autocarros da STCP (2018), no centro da cidade do Porto - Portugal, o PIS empregado é responsável por exibir em um painel eletrônico junto à paragem, o tempo estimado para a chegada dos próximos autocarros. Nos autocarros da STCP (2018) há um PIS responsável por exibir no conjunto de painéis eletrônicos espalhados pela parte frontal, lateral e interior do autocarro, informações acerca da linha e do destino do autocarro. Este último PIS é o alvo central deste trabalho.

Os PIS responsáveis por indicar linha e destino do autocarro ao qual estão instalados, são conhecidos como Indicadores de Destino. Os Indicadores de destino podem ser utilizados em diferentes modelos de autocarros, tanto para urbanos como para os de longos trajetos (DMS, 2004). É possível encontrar no mercado diversos fabricantes, com diferentes modelos de indicadores de destino.

A DMS (*Display & Mobility Solutions*), é uma empresa localizada na cidade

do Porto, em Portugal, subsidiária da *Microprocessador (Digital Systems)*. O qual possui mais de 30 anos no mercado de sistemas para informações públicas, painéis eletrônicos e soluções de segurança e mobilidade. A DMS possui experiência na implantação de sistemas de informações públicas em mais de 10 países e uma linha de indicadores de destino versáteis e seguros, já bem consolidados no mercado (DMS, 2018). Segundo a DMS (2018), o *SICON II*, é um dos modelos indicador de destino mais vendidos, seu projeto datado de 2004 vem sofrendo gradativas atualizações ao longo dos anos. Este modelo de indicador de destino será o modelo de referência neste trabalho.

Com o crescimento do sistema de transporte público por meio de autocarros, a modernização das frotas utilizadas e o desenvolvimento de novas tecnologias de sistemas embarcados, a DMS constatou a necessidade de também modernizar a sua linha de indicadores de destino. De modo a fornecer aos seus clientes um sistema mais versátil, inteligente, confiável e barato (DMS, 2018). O console, ou módulo de interface com o utilizado, dentro do sistema dos indicadores de destino é um elemento chave, já que este coordena toda a rede de painéis eletrônicos no autocarro, sendo também mais suscetível a modernização. Neste trabalho pretende-se desenvolver um novo modelo de console, que atenda as necessidades de modernização e renove a linha de indicadores de destino da DMS.

## 1.1 ESTRUTURA DO TRABALHO

A fim de apresentar de forma clara e organizada o caminho pelo qual este trabalho atingiu seus objetivos, a divisão dos capítulos fora feita em 4 partes; Revisão de Literatura, Desenvolvimento da Aplicação DMSConsole, Montagem e Teste do Protótipo, e por fim a Conclusão. No capítulo (2) são apresentados os principais componentes e conceitos envolvidos no desenvolvimento do novo modelo de console, como o protocolo de comunicação dos painéis eletrônicos, sistema de indicadores de destino e desenvolvimento de aplicações no Raspberry Pi. No capítulo (3) é demonstrado o desenvolvimento da aplicação DMSConsole para Raspberry Pi, responsável pela gestão da rede de painéis e da interface do usuário. No capítulo (4) é onde efetivamente o protótipo de console é montado, sendo aqui desenvolvido também a placa de circuito impresso utilizada para a integração dos componentes do projeto. Ao fim o capítulo (5) apresenta a conclusão deste trabalho. Por opção do autor, este trabalho foi escrito em português brasileiro, que corresponde à sua língua materna. A grafia utilizada enquadra-se no acordo ortográfico vigente na data de elaboração deste

trabalho.

## 1.2 RECURSO UTILIZADOS

Para realizar o desenvolvimento do novo console para indicadores de destino, a empresa DMS disponibilizou todos os componentes de *hardware* e os *software* necessários. Os principais recursos utilizados são:

- Raspberry Pi - Zero W
- *Display Touchscreen* TOPWAY - HKT080ATA-C
- Conversor UART TTL/UART RS232 TI-MAX3232
- Conversor UART TTL/UART RS485 TI-SN65HVD3082ED
- Conversor CC-CC Chaveado OKI-78SR
- Painéis eletrônicos de indicadores de destino
- *Software Altium PCB Design*

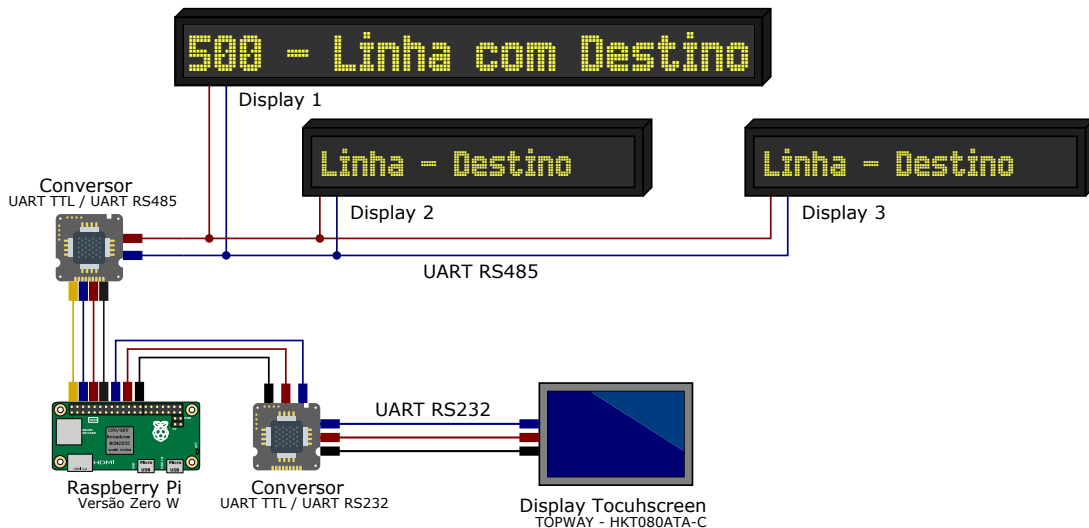
O *Raspberry Pi - Zero W* é *hardware* utilizado para desenvolver a aplicação. Já o o *Display Touchscreen* TOPWAY - HKT080ATA-C é o periférico usado como interface do usuário. O conversor UART-RS485 é utilizado na comunicação entre o *Raspberry Pi* e a rede de painéis eletrônicos, e o UART-RS232 é utilizado na comunicação entre o *display touchscreen* e o *Raspberry Pi*. O conversor OKI-78SR é utilizado como fonte de alimentação para o console. O *Software Altium PCB Design* é um CAD (*Computer Aided Design*) específico para projeto eletrônicos, especialmente *design* de placas de circuito impresso.

## 1.3 OBJETIVOS

### 1.3.1 OBJETIVOS GERAIS

O objetivo principal desde trabalho é o desenvolvimento de um novo modelo de console, para a linha de Indicadores de Destino fabricados pela empresa DMS. Tal console deve ser capaz de se comunicar com o mesmo protocolo utilizado pelo antigo console presente no modelo de indicadores de destino *SICON II*. Porém este novo modelo deve possuir um *display touchscreen* de 7", através do qual deve apresentar uma

interface de usuário, ou *User Interface* (UI), amigável e simplificada. O novo modelo deve apresentar recursos como; conexão *Wi-Fi*, carregamento de ficheiros de destino via USB, e acesso remoto seguro. Um *hardware* hábil a atualizações de software e implementação de novas funções de acordo com as necessidades de cada cliente é uma requisito deste novo modelo de console. A Figura (1) apresenta o diagrama de projeto para o novo console para indicadores de destino.



**Figura 1: Diagrama do Sistema de Indicadores de Destino para o Novo Console - DMS**  
**Fonte: Autoria Própria**

### 1.3.2 OBJETIVOS ESPECÍFICOS

Para o novo modelo de console é necessário desenvolver um conjunto de ferramentas em *software*, capazes de realizar a comunicação e o controle dos devidos periféricos, e ainda gerir os dados e a interface de usuário. Por outro lado é importante também utilizar um conjunto de *hardware* capaz de atender as necessidades, e que torne o desenvolvimento ágil. Assim os objetivos específicos deste trabalho são:

- Desenvolver a ferramenta em *software* para realizar a comunicação entre o periférico de interface de usuário *Display Touchscreen* TOPWAY e o *Raspberry Pi*.
- Desenvolver a ferramenta em *software* para a comunicação entre a rede de painéis eletrônicos e o *Raspberry Pi*, utilizando o protocolo de comunicação padrão DMS.
- Desenvolver a ferramenta em *software* para a gestão do dados referente as informações a serem mostradas nos painéis. Permitindo ao usuário inserir, remover e manipular tais dados dentro do sistema.

- Desenvolver a aplicação que englobe as ferramentas de comunicação e gestão de dados, e que ainda faça o controle e configuração de todos o sistema.
- Montagem do protótipo de novo modelo de console para os indicadores de destino.



## 2 TECNOLOGIAS ENVOLVIDAS

### 2.1 DISPLAY TOUCHSCREEN TOPWAY - HKT080ATA-C

A utilização de um ecrã sensível ao toque, ou *Display touchscreen*, no projeto do console de indicadores de destino tem como objetivo tornar a interface do usuário mais versátil e interativa. Utilizando um *Display touchscreen* é possível criar um conjunto de regras que delimitam certas zonas da superfície do *Display*, que ao serem tocadas pelo usuário ativam o envio de mensagens ao dispositivo mestre, informando a ocorrência do evento. Utilizando este recurso é possível criar botões virtuais no *Display* que desencadeiam eventos no dispositivo mestre, o qual de acordo com a sua programação responde ao *display* com requisições que podem mudar os aspectos visuais do *frame* exibido. Através da dinâmica de botões virtuais e da mudanças de *frames* torna-se prático a criação de uma interface de usuário que se molda, criando botões e recursos visuais de acordo com a necessidade das requisições do usuário. O TOPWAY HKT080ATA-C, apresentado na Figura (2) é o modelo de *Display touchscreen* disponibilizado pela DMS, para a realização deste trabalho.

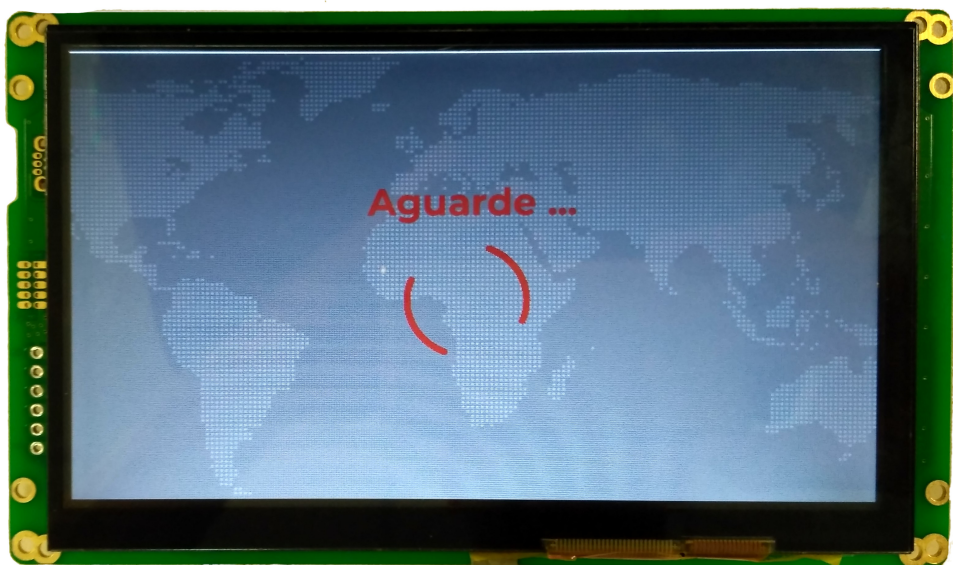
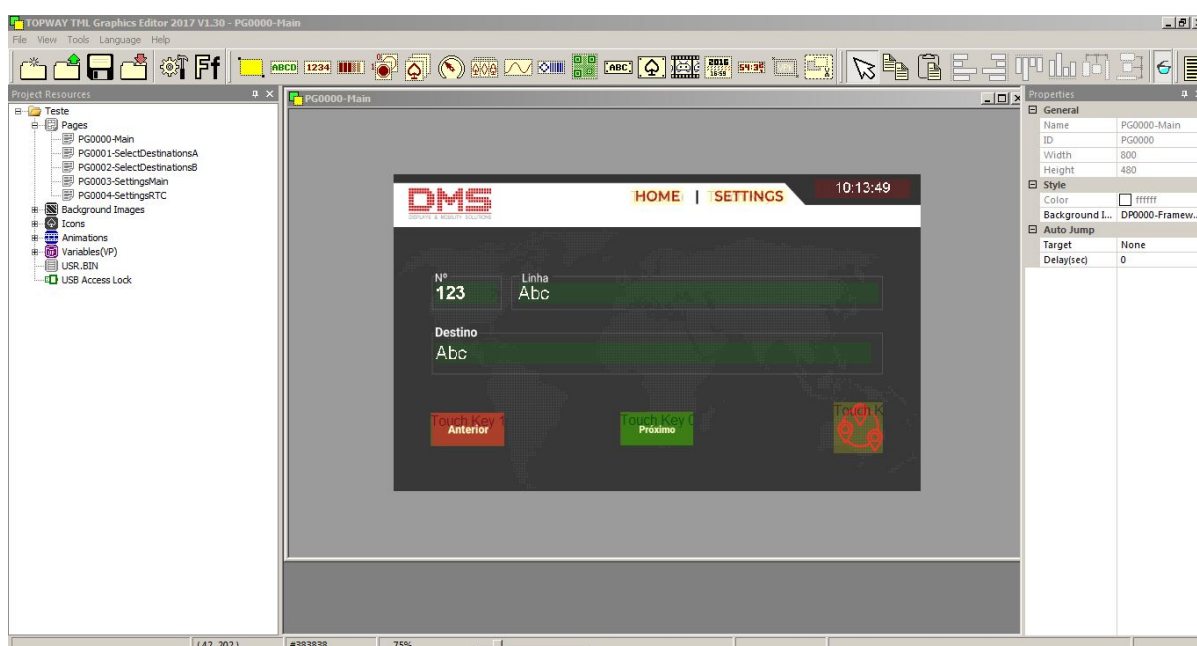


Figura 2: *Display Touchscreen* TOPWAY HKT080ATA-C  
Fonte: Autoria Própria

O TOPWAY HKT080ATA-C é um módulo *Smart display* de 7 polegadas TFT LCD (*thin film transistor liquid crystal display*), o qual possui uma interface de comunicação UART-RS232. Por ser um módulo *Smart* este *display* vem equipado com um microcontrolador que gerencia os arquivos de imagens, armazenados na memória RAM também presente no módulo, com os quais ele constrói os *frames* a serem exibidos. Através da comunicação UART o dispositivo mestre configura e controla o microcontrolador.

O fabricante do TOPWAY HKT080ATA-C disponibiliza um programa de edição gráfica de interface, o *TOPWAY TML Graphics Editor*, visto na Figura (3). Este editor cria o conjunto de dados a serem exportados, via conexão USB, para dentro da memória RAM do módulo.



**Figura 3: Programa de Edição de Interface TOPWAY TML Graphics Editor**  
**Fonte: Autoria Própria**

Por meio desta aplicação é possível instanciar elementos gráficos, configurar comportamentos responsivos, alocar variáveis que armazenam eventos e configurar gatilhos para transmitir via UART-RS232 as informações sobre eventos. Ainda é possível designar áreas de texto na interface, que são preenchidas com informações que podem ser enviadas de um dispositivo mestre para o *display* via UART. Assim basta criar os elementos gráficos da interface e suas ações, através do *TOPWAY TML Graphics Editor*, e gerenciar a interface usando o dispositivo mestre. O dispositivo

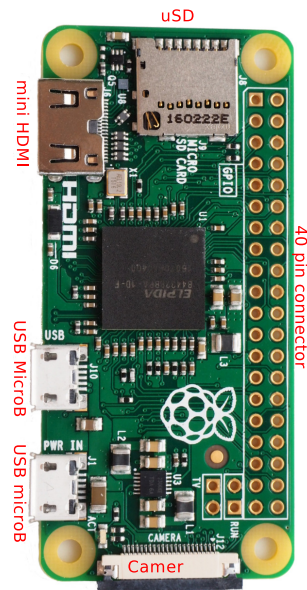
mestre utilizado neste trabalho é o Raspberry Pi - Zero W.

## 2.2 RASPBERRY PI ZERO W

Raspberry Pi é uma serie de computadores de tamanho reduzido construídos em uma única placa de circuito impresso (*single-board computers*). Desenvolvido pela Raspberry Pi Foundation, os modelos da série Raspberry Pi possuem um Broadcom SoC (*System on a Chip*) integrado com um processador ARM e uma unidade de processamento gráfico (GPU) on-board (ADAFRUIT, 2018). Mesmo sendo criado com o intuito de promover ensino de ciências da computação em escolas de ensino básico, o Raspberry Pi tem sido encontrado em diversas aplicações na bibliografia por conta de sua versatilidade, preço baixo e tamanho reduzido (TSO *et al.*, 2013),(PRIYANKA *et al.*, 2017).

Em fevereiro de 2017, a Raspberry Pi Foundation lançou um novo modelo de Raspberry Pi no mercado, o Raspberry Pi Zero W, visto na Figura (4). Este novo modelo possui a metade do tamanho do modelo Raspberry Pi 3 A+, porém com o dobro de recursos e conectividade Wi-Fi. O Pi Zero W possui um processador ARM de 1GHz, memória RAM de 512 MB, porta USB OTG (*On-The-Go*), porta Mini HDMI, conector para camera CSI e 40 pinos de GPIO (*General Purpose Input/Output*) (ADAFRUIT, 2018).

O Raspberry Pi Zero W, assim como todos os modelos Raspberry Pi, suportam sistemas operacionais baseados no kernel Linux, dispondo de toda as bibliotecas básicas destes tipos de sistemas. A Raspberry Pi Foundation disponibiliza uma versão de Linux, baseada na distribuição Debian (DEBIAN, 2005), chamada Raspbian. Raspbian foi desenvolvido com o intuito de otimizar os recursos computacionais disponíveis no hardware, permitindo rodar uma interface gráfica amigável e ainda manter um bom desempenho. Em sua versão inicial, em 2012, o Raspbian possuía mais de 35000 pacotes de aplicações otimizadas para os modelos Raspberry Pi. Porém como este é um sistema Linux, é possível instalar pacotes de outras distribuições apenas incluindo e instalando as devidas dependências (DEBIAN, 2005). O Raspbian também possui compilador para as linguagens como Java, C/C++ e especialmente Python (ADAFRUIT, 2018).



**Figura 4: Raspberry Pi Zero W**  
65mm de comprimento x 30 mm de lar-  
gura x 5mm de espessura  
Fonte: Sparkfun (2018)

O Raspberry além de oferecer um dispositivo poderoso com um sistema operacional Linux, ainda agrega as funcionalidades dos microcontroladores mais comuns em sistemas embarcados. Entre estas funcionalidades está o conjunto de pinos da GPIO (*General Propose Input and Output*), o qual é uma das grandes responsáveis pela facilidade com que o Raspberry Pi Zero W se encaixe nas mais diversas aplicações. O sistema operacional, no Raspberry Pi Zero W, tem acesso a 28 portas digitais de entrada/saída da GPIO, por onde é possível realizar nos pinos da GPIO processos de leitura e escrita de sinais digitais, PWM (*Pulse-Width Modulation*) e estabelecer comunicação pelos protocolos UART, I2C e SPI.

		3.3V	1	2	5V		
I2C	SDA	GPIO2	3	4	5V		
	SCL	GPIO3	5	6	GND		
	GPCLK0	GPIO4	7	8	GPIO14	Tx	UART
		GND	9	10	GPIO15	Rx	
		GPIO17	11	12	GPIO18	PWM0	
		GPIO27	13	14	GND		
		GPIO22	15	16	GPIO23		
		3.3V	17	18	GPIO24		
SPI	MOSI	GPIO10	19	20	GND		
	MISO	GPIO9	21	22	GPIO25		
	SCLK	GPIO11	23	24	GPIO8	CS0	SPI
	GND	25	26	GPIO7	CS1		
		DNC	27	28	DNC		
	GPCLK1	GPIO5	29	30	GND		
	GPCLK2	GPIO6	31	32	GPIO12		
	PWM1	GPIO13	33	34	GND		
	PWM1	GPIO19	35	36	GPIO16		
		GPIO26	37	38	GPIO20		
		GND	39	40	GPIO21		

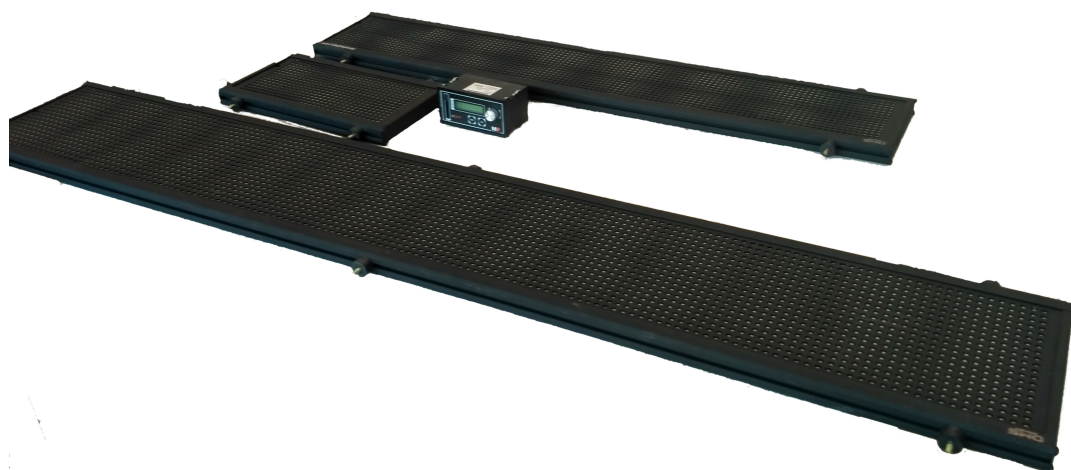
**Tabela 1: Configuração GPIO - Raspberry Pi Zero**

Fonte: (ADAFRUIT, 2018)

A Tabela (1) apresenta um resumo dos recursos presentes na GPIO do Raspberry Pi Zero W. Nesta tabela as portas que possuem o identificador GPIOx representam um pino da GPIO que pode ser usada para a leitura ou escrita de dados digitais. Já as portas marcadas com SPI, UART ou I2C são aquelas que suportam leitura e escrita de dados digitais, mas que podem ser configuradas exclusivamente como interfaces de comunicação. Existem ainda os pinos de PWM, os quais podem ser usados para a modulação, os pinos de alimentação 3.3/5V e GND (Terra), e ainda os pinos reservados DNC (*Do Not Connect*). É importante notar que no Raspberry Pi Zero W existe apenas uma interface de comunicação UART disponível, porém no projeto alvo deste trabalho existe a necessidade de duas interfaces UART. Para resolver este problema será utilizado um recurso de software para tornar alguns das portas da GPIO hábeis para a comunicação UART.

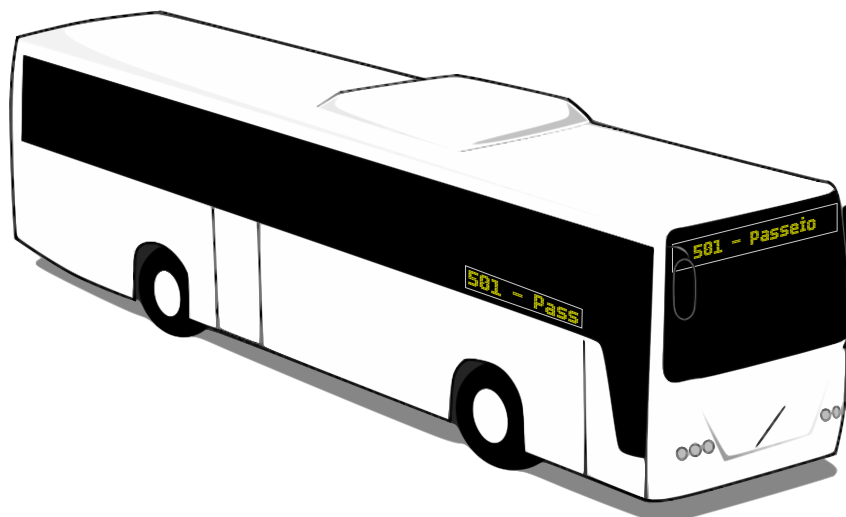
### 2.3 INDICADORES DE DESTINO - DMS

Os indicadores de destino são recursos visuais de identificação, utilizados em autocarros de diferentes tipos. O principal objetivo dos indicadores é informar ao público acerca do trajeto e/ou do destino do autocarro. Os indicadores podem ser analógicos, constituídos basicamente de um sistema mecânico, eletromecânico ou ainda totalmente eletrônico. A Figura (5) apresenta o indicador eletrônico de destino *SICON II* da DMS. Os dados técnicos do *SICON II* foram disponibilizado pela DMS, pra a realização deste trabalho.



**Figura 5: Indicador de Destino *SICON II* - DMS**  
**Fonte: DMS (2018)**

É comum haver mais de um painel eletrônico em um autocarro, de modo a tornar visível de qualquer lado de uma via as informações dos painéis. Os painéis são comumente distribuídos no autocarro pelo para-brisa e pelas janelas laterais, como pode ser visto na Figura (6).



**Figura 6: Autocarro Modelo - 2 Painéis Eletrônicos de Indicação de Destino**  
**Fonte: Autoria Própria**

Para que seja possível controlar todos os painéis de forma centralizada é estabelecido uma rede de comunicação entre os painéis e um console de controle. O sistema de indicadores de destino é composto basicamente pela rede de painéis eletrônicos e pelo console, como mostrado na figura (7).



**Figura 7: Diagrama do Sistema de Indicadores de Destino - DMS**  
**Fonte: Autoria Própria**

O console controla os painéis via UART-RS485, recebendo informações de estados dos painéis, e enviando as configuração e os dados a serem exibidos. É por meio do console que o usuário insere, utilizando a conexão USB, os conjuntos de dados fonte referente a textos que devem ser exibidos nos painéis. A Figura (8) apresenta o console do Indicador de Destino *SICON II* da DMS.

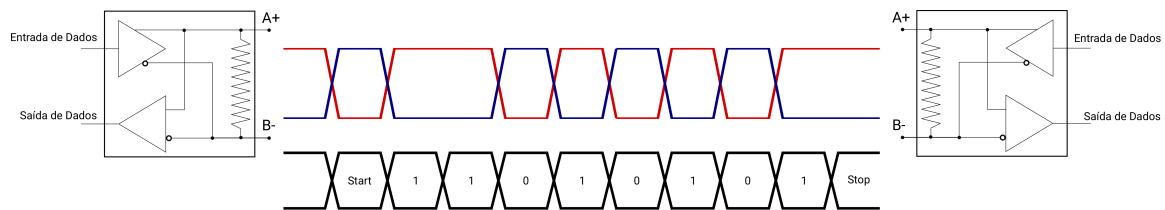


**Figura 8: Console do Indicador de Destino SICON II - DMS**  
Fonte: DMS (2018)

## 2.4 INTERFACE DE COMUNICAÇÃO UART-RS485

Os painéis eletrônicos utilizados nos indicadores de destino da DMS possuem protocolos de comunicação próprios, através interface de comunicação UART-RS485. Este padrão é muito utilizado no ambiente industrial, e em aplicações onde o meio de comunicação está exposto a interferências magnéticas diversas, como é o caso dos autocarros.

UART RS-485, ou *UART Recommended Standard-485* é uma interface de comunicação especificada pela norma TIA/EIA-485, e aprovada em 1983 pela *Electronics Industries Association* como uma nova interface de transmissão de dados por linhas diferenciais, capaz de se comunicar com 32 unidades de carga. Nesta interface o meio de comunicação é um par transado, por onde cada dispositivo transmite e recebe informações. O par transado nada mais é do que um meio de transmissão diferencial, ou seja quando um bit '1' é transmitido a diferença de potencial entre o terminal nomeado positivo (ou A+) é maior do que o terminal negativo (ou B-), já quando o bit '0' é transmitido a diferença de potência entre o terminal positivo é menor que o terminal negativo (TEXAS, 2008). Como pode ser visto na Figura (9).

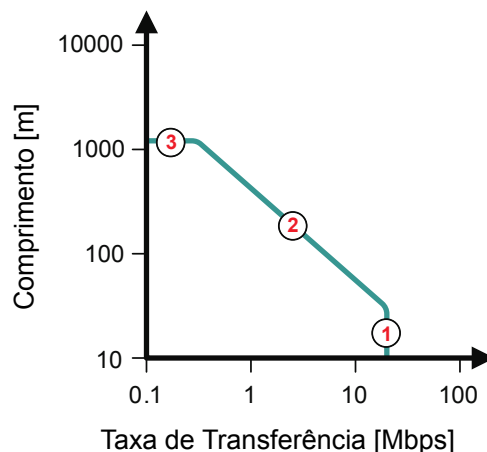


**Figura 9: Interface de Comunicação UART RS-485**

Fonte: Texas (2008)

Segundo a Texas (2008) a utilização do UART RS485 resulta no cancelamento de ruídos induzidos no meio da transmissão. Já que qualquer ruído induzido por campos magnéticos interfere nos 2 condutores simultaneamente. Quando o amplificador diferencial, presente nos conversores, recebe os dois sinais a diferença de tensão não se altera e o ruídos induzidos nos dois condutores são eliminados. Como esta interface de comunicação se estabelece apenas pela diferença de potencial entre os dois condutores, o condutor terra é dispensado e o nível de tensão usado nesta comunicação é mais flexível. O transmissor de RS-485 deve oferecer no mínimo 1,5/-1,5 V na comunicação, já o receptor deve possuir uma sensibilidade de no mínimo 200mV/-200mv. A norma TIA/EIA-485 ainda especifica que a diferença de potencial máxima entre equipamentos é 12/-7 V, para que a tensão elevado no par trançado não danifique os equipamentos.

Como o padrão 485 possui uma tensão elevada, e um limite mínimo de tensão de recepção em 200mV/-200mv, o efeito da queda de tensão nos condutores é mais tolerável. Assim segundo a norma a distância máxima da rede UART-485 é de 1.219 metros. Em relação a velocidade máxima da rede, esta é definida de acordo com o comprimento dos condutores, como mostrado no gráfico da Figure (10) (TEXAS, 2008).



**Figura 10: Comprimento da Rede Vs Taxa de Transferência**

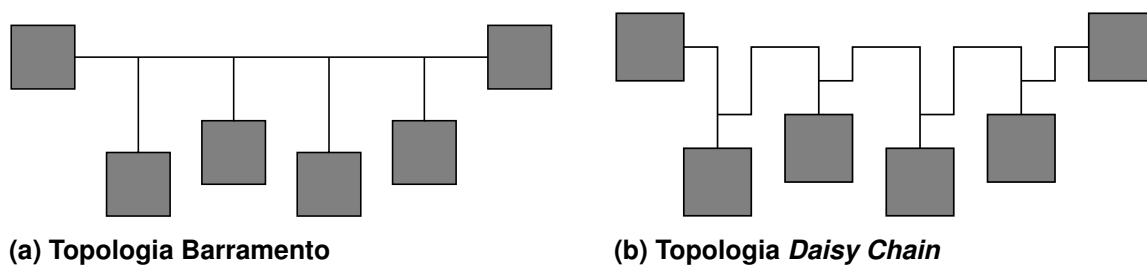
(1) Alta velocidade de transmissão:  $\leq 12$  Mbps

(2) Velocidade média de transmissão:  $1 \text{ Mbps} \leq \leq 10$  Mbps

(3) Baixa velocidade:  $\leq 200$  kbps

Fonte: Adaptado de Texas (2008)

A topologia a ser utilizada no RS-485 é a de barramento ou *Daisy Chain*, o que permite a adição ou remoção de elementos sem que isso influencie os demais dispositivos já conectados (TEXAS, 2008), como mostrado na Figura (11). A norma TIA/EIA-485 define como limite 32 unidades de cargas conectadas, porém existem dispositivos que consomem 1/4 ou até 1/8 de uma unidade de carga, o que na prática expande os limites de dispositivos conectados.



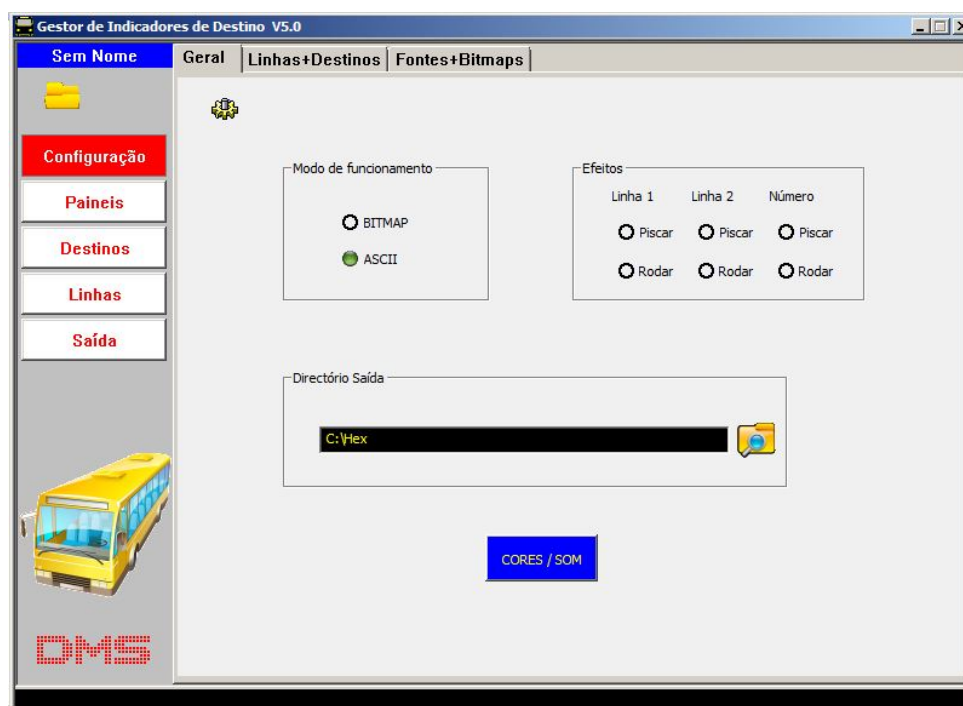
**Figura 11: Tipos de Topologia para UART-RS485**

Fonte: (TEXAS, 2008)

## 2.5 GESTOR DE INDICAÇÃO DE DESTINOS

Os dados acerca das informações a serem exibidas nos painéis eletrônicos dos indicadores de destino do modelo SICON II, são arquivos binários carregados

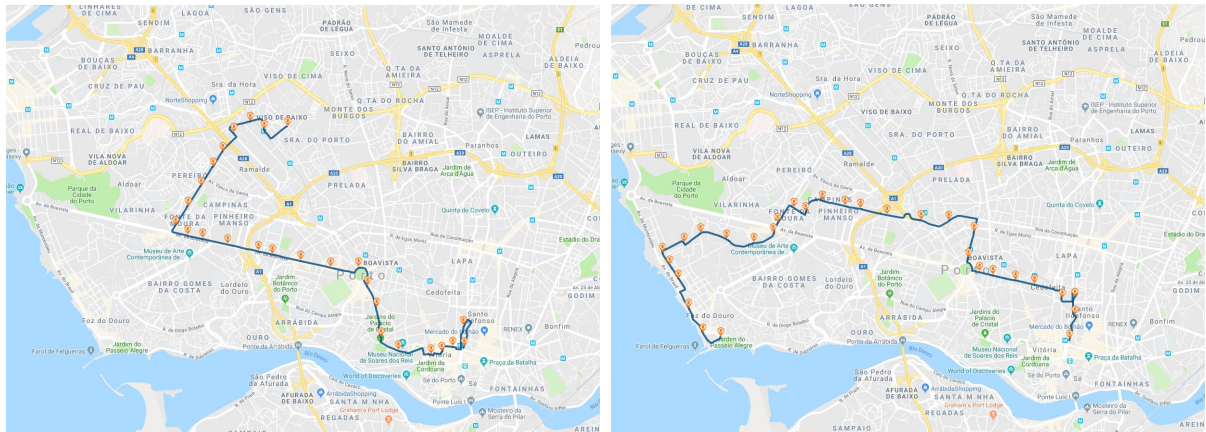
para dentro da memória interna do console através de uma conexão USB. Tais dados são gerados pela aplicação GID, ou Gestor de Indicadores de Destinos, fornecido pela DMS, Figura (12). Por meio desta aplicação o cliente pode gerir os textos a serem usados nos indicadores de destino, podendo configurar fontes e efeitos em cada painel independentemente.



**Figura 12: Gestor de Indicadores de Destino - Tela Inicial**  
Fonte: DMS (2004)

As informações exibidas em um painel eletrônico de um sistema de indicadores de destino são, geralmente, relativas ao itinerário atual do autocarro. A estrutura de dados que o GID gera esta relacionado com a definição de linha e destino do autocarro. A linha é a denominação do percurso atual do autocarro, já o destino é referente a próxima paragem do autocarro ou, como acontece nos autocarros da STCP, o último ponto de paragem da linha. Por exemplo, nas Figuras (13a) e (13b) estão marcados respectivamente os trajetos das linhas 201 e 202 da STCP. Quando um autocarro realiza um trajeto partindo da paragem 'Avenida dos Aliados', passando por todas as paragens até a 'Viso de Baixo', os painéis eletrônicos devem exibir a Linha '201', e o destino 'Viso de Baixo'. Porém no retorno a 'Avenida dos Aliados' a Linha continua sendo a '201', e o destino passa a ser a 'Avenida dos Aliados'. Já o autocarro que realiza o trajeto partindo da 'Avenida dos Aliados' até 'Passeio Alegre', os seus painéis

devem exibir a Linha 202, com o destino a 'Passeio Alegre', e no retorno conserva-se a Linha porém o destino passa a ser a 'Avenida dos Aliados'.



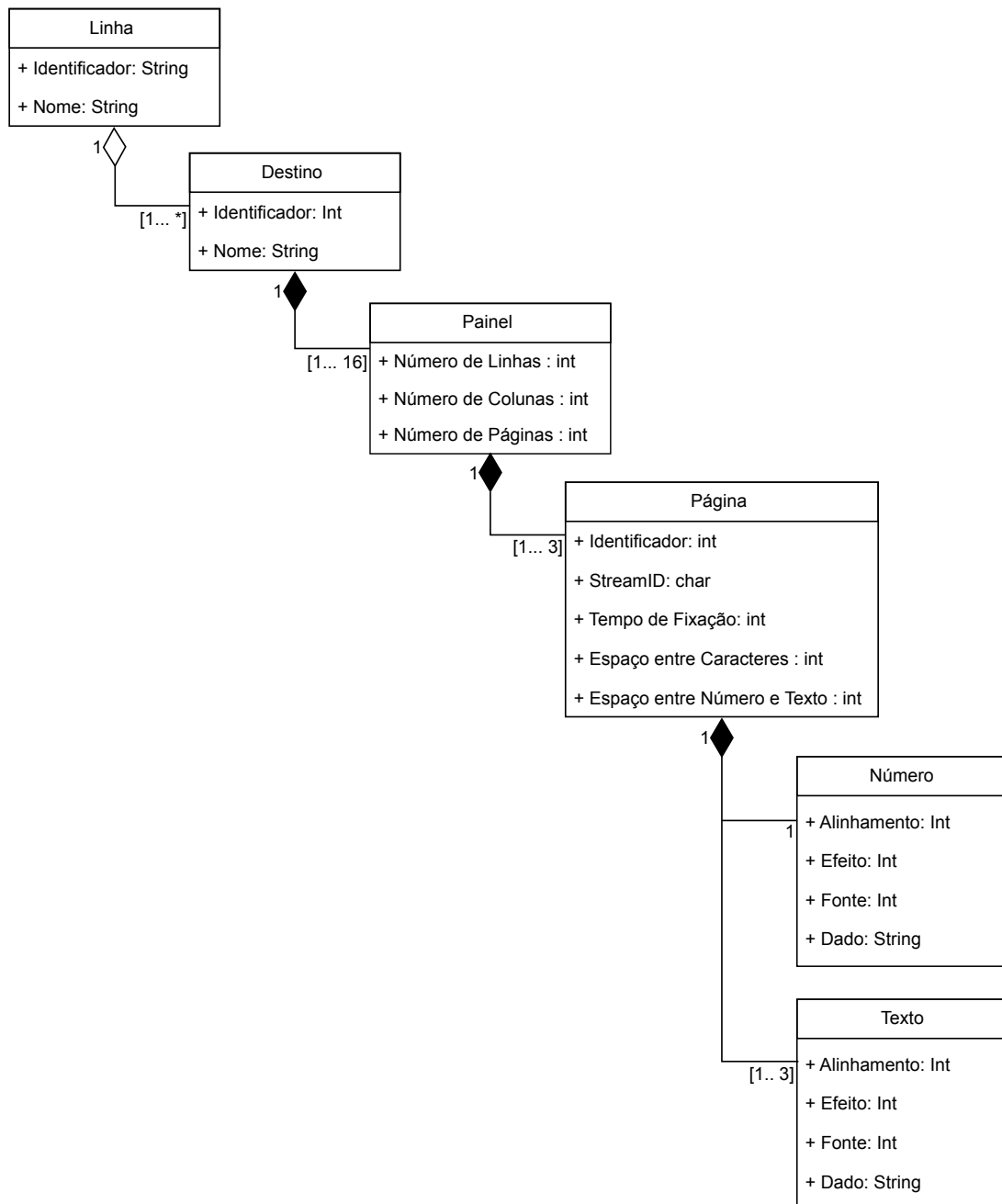
(a) Linha 201, Avenida dos Aliados - Viso de Baixo

(b) Linha 202, Avenida dos Aliados - Passeio Alegre (Via AV. Bessa)

**Figura 13: Linhas de Autocarro da STCP**  
Fonte: STCP (2018)

Dentro da estrutura de Dados utilizados no GID há 5 classes de objetos; 'Linha', 'Destino', 'Painel', 'Página' e 'Campo'. O objeto 'Linha' representa o percurso total realizado pelo autocarro através das várias paragem. Este objeto possui como atributos o nome e identificador da Linha, e uma lista de  $n$  objetos 'Destino', que representam as possíveis paragens finais da linha. Tomando como exemplo a linha 201, o seu dado objeto tem como atributo identificador '201', nome 'Aliados-Viso', e a lista de destinos constituído de dois elementos; 'Avenida dos Aliados' e 'Viso'. O objeto 'Destino' possui atributo nome e identificador do destino, e uma lista de até 16 elementos do tipo 'Painel'.

Um elemento do tipo 'Painel' agrega um conjunto de 'Páginas', que descrevem a informação visual de um 'Destino' num formato adequado para um dado painel eletrónico. Para além dos textos a exibir, cada 'Pagina' define também parâmetros de espaçamento, tempo de afixação, fontes e outros efeitos visuais. Esta estrutura permite que o mesmo destino seja apresentado de forma textual num painel frontal de grandes dimensões, enquanto num painel lateral reduzido o mesmo destino é apresentado apenas com o número da linha. A Figura 14 apresenta esta estrutura de dados.



**Figura 14: Estrutura de Dados do GID**  
 Fonte: DMS (2004)

Na aplicação GID existe uma tela nomeada 'Destinos', onde o usuário insere a lista de destinos possíveis, e atribui um conjunto de destinos a uma 'Linha' específica, como mostrado na Figura (15a). Por meio da tela nomeada 'Linhas', como visto na Figura (15b), o usuário insere a lista de 'Linhas' disponíveis e atribui um grupo de painéis para cada 'Linha'. O grupo de painéis eletrônico disponível na rede do indicador de destino é configurado pelo usuário na tela 'Painéis'. Ao final o usuário utiliza a tela 'Saída' para ajustar as fontes utilizada nos textos e números exibidos

nos painéis, e ainda adiciona ou remove páginas e campos em cada um dos painéis atribuídos aos destinos, vide Figura (15d).



**Figura 15: Gestor de Indicadores de Destino**  
Fonte: DMS (2004)

Se todas as configurações escolhidas na tela 'Saída' estiverem coerentes com as dimensões e capacidades dos painéis, ao escolher a opção 'Gerar Fichero de Dados', o GID gera um conjunto de arquivos binários, que devem ser movidos para uma unidade de armazenamento e conectada ao console do Indicador de Destinos. Ao conectar a unidade de armazenamento a porta serial do console, os dados são movidos para a memória interna do console, e os dados binários são enviados aos painéis.

### 2.5.1 PROTOCOLO LDESIGN

Os painéis eletrônicos utilizados nos indicadores de destino da DMS possuem um protocolo de comunicação próprio desenvolvido pela DMS, o qual é chamado

de protocolo LDESIGN. Este protocolo possibilita que seja enviada informação de controle e notificação para qualquer dispositivo que integra a rede de comunicação dos indicadores. A rede de comunicação dos indicadores de destino é normalmente composta por um console, que funciona no modo *Master*, e até 16 painéis funcionando do modo *Slave*. No protocolo de comunicação LDESIGN qualquer mensagem enviada ou recebida por um dispositivo da rede segue o seguinte formato:

Ordem	Comando	Descrição	Nº bits
1º	<SOH>	<i>Start of heading</i> - 81h	8
2º	Destino	ID do dispositivo destino	8
3º	Origem	ID do dispositivo origem	8
4º	<STX>	<i>Start of text</i> - 82h	16
5º	N	Tamanho em Bytes dos Dados + 1	8
6º	Tipo	Tipo de dado enviado	8
7º	Dados	Dados a serem enviados	N
8º	<ETX>	<i>End of Text</i> - 83h	8
9º	<CHKS>	<i>checksum</i> -soma de todos os Bytes da mensagem, SOH até ETX	8

**Tabela 2: Formato Geral - Protocolo Geral de Comunicação LDESIGN**

**Fonte: DMS (2004)**

O ID (*Identificador*) de destino e de origem são dados específicos de cada dispositivo. Normalmente o console tem por padrão o ID 0x00 e o painel com as maiores dimensões da rede tem o ID 0x01. Quando o console faz uma requisição a um painel específico, por exemplo ao painel ID 0x05, o campo de 'Destino' e 'Origem' são respectivamente 0x05 e 0x00, e o painel responde com uma mensagem com ID de Destino 0x00 e Origem 0x05. Porém quando o console deseja realizar uma requisição tipo *Broadcast*, ou seja a todos os painéis da rede ao mesmo tempo, o ID de Destino deve ser 0xFF e a Origem 0x00. Cada painel irá responder a requisição do console de forma sequencial, cronometrada de acordo com seu valor de ID multiplicada por 100 ms. Por exemplo, em uma rede com dois painéis de ID 0x01 e 0x03, o painel de ID 0x01 responde 100 ms depois do fim da transmissão *Broadcast* do console. Já o painel de ID 0x03 responde 300 ms depois do fim da da transmissão *Broadcast* do console, garantindo assim a não sobreposição de respostas dos painéis.

O campo 'Tipo' identifica o tipo de mensagem a ser enviada, enquanto o

campos 'Dados' a descreve. Existem 7 tipos diferentes de mensagens dentro do protocolo de comunicação LDESIGN, cada uma com funções e sintaxe de 'Dados' diferentes. Para evitar a redundância de informações neste trabalho, será na seção (3.2.8) apresentado os tipos de mensagem dentro do LDESIGN, e conseqüentemente definido os valores utilizados nos campos 'Tipo' e 'Dados'.

### 3 DESENVOLVIMENTO DA APLICAÇÃO DMSCONSOLE

Nesta etapa do trabalho é desenvolvido o *software* implementado no Raspberry Pi Zero W, para o novo modelo de console de indicadores de destino da DMS. Este *software* será aqui denominado DMSConsole versão 1.0, e será basicamente desenvolvido na linguagem de programação C++, utilizando algumas bibliotecas nativas de sistemas operacionais que utilizam o Kernel Linux e outras que são específicas para *hardware* Raspberry Pi. Portanto o *software* desenvolvido aqui pode ser compilado em diferentes modelos de *hardware* Raspberry Pi, apenas necessitando de pequenos ajustes nas portas utilizadas.

O desempenho do DMSConsole é um elemento crucial nesta aplicação, pois o tempo de execução de tarefas do *software* pode interferir na execução das rotinas de comunicação dos painéis, comprometendo o correto funcionamento dos mesmo. O tempo de resposta á requisições feitas pelo usuário através do *Display Touchscreen* e o tempo de inicialização do console são parâmetros que podem comprometer a experiência do usuário, e de mesmo modo dependem do desempenho da aplicação nas diferentes tarefas executadas. Assim utilizando os recursos que a linguagem C++ oferece, o desenvolvimento de cada função da aplicação DMSConsole é feita visando a otimização de recursos computacionais.

Para que o aplicação DMSConsole inicie automaticamente com o sistema Linux presente no Raspberry Pi, o script de instalação desta aplicação cria um script de inicialização dentro do diretório `/etc/init.d/` do Linux, no padrão LSB (*Linux Standard Base*). Assim o DMSConsole inicia juntamente com o sistema, ficando em *background*, respondendo aos comandos de *start*, *stop* e *restart* padrão de serviços Linux.

#### 3.1 GERENCIADOR XML

Como foi visto na Seção (2.5) a aplicação GID a partir das informações inseridas pelo usuário, gera um conjunto de arquivos binários com as informações a serem inseridas no console, e conseqüentemente enviadas aos painéis eletrônicos. Este conjunto de dados gerados possui uma sintaxe própria, que somente a aplicação GID gera e somente o console do Indicador de Destino da DMS consegue ler e iden-

tificar a estrutura dos dados. Com a intenção de tornar o arquivo fonte de dados para Indicadores de Destino mais padronizado e portátil para diferentes versões de console, será utilizado o XML como padrão de descrição de dados. Assim será possível gerar fonte de dados, para os Indicadores de Destino, através de qualquer aplicação de criação de arquivos XML, utilizando apenas a estrutura de dados apresentada na Figura (14).

O XML, ou Linguagem de Marcação Extensível (*Extensible Markup Language*), é um formato para criação de dados organizados de forma hierárquica, recomendado pelo W3C (*World Wide Web Consortium*). O XML é uma combinação entre a flexibilidade que SGML, ou *Standard Generalized Markup Language* e a simplicidade visual que o HTML, ou *HyperText Markup Language*, oferecem. Assim o XML oferece: simplicidade e legibilidade dos dados, tanto para usuários quanto para *softwares*; Flexibilidade, possibilitando a criação de *tags* ilimitadas; interligação facilitada entre banco de dados e uma boa estruturação (BRAY *et al.*, 1997).

A estrutura de dados do GID, apresentada na Figura (14), é constituída por *n* elementos 'Linha', que serão designados no padrão XML como 'Line', que estão contidos dentro de um conjunto de dados fontes para os indicadores de destino. Como padrão da DMS, todos conjuntos de dados de uma 'Linha' deve estar associada a um conjunto de painéis, e este conjunto associado ao elemento raiz 'GID'. Assim é criado um elemento 'GID' (Gestor de Indicadores de Destino), o qual pode possuir *n* elementos 'Group', os quais representam grupos de painéis eletrônicos que estão a ser geridos. Estes grupos contém os elementos 'Line', com seus atributos e seus elementos relacionados.

Logo como exemplo, considerando um arquivo XML que contém informação a cerca de uma linha de autocarros denominada 'Aliados-Viso' de número 201, o qual possui como destinos possíveis o 'Planetário' ou a rua 'PR. D. JOÃO I'. E que exista apenas um painel de dimensões 16x112 pixels, onde as informações devam ser mostradas em uma página de configuração número da linha + nome do destino, com espaço de 2 pixels entre o número e nome, e com 1 pixel entre caracteres do nome. O arquivo XML para este GID específico é dado da seguinte forma:

```
1 <GID version="1.1">
2   <Group ID="1">
3     <Line ID="201" Name="ALIADOS-VISO">
4       <Destination ID="1" Name="PLANETÁRIO">
5         <Panel Lines="16" Columns="112">
6           <Page ID="1" SBC="1" SBNAncC="2" PT="15">
```

```
7      <Number Effect="None" Align="Left" Font="16x9">
8          201
9      </Number>
10     <Text Effect="None" Align="Left" Font="5x5">
11         PLANETÁRIO
12     </Text>
13 </Page>
14 </Panel>
15 </Destination>
16 <Destination ID="2" Name="PR. D. JOÃO I">
17     <Panel Lines="16" Columns="112">
18         <Page ID="6" SBC="1" SBNAnc="2" PT="15">
19             <Number Effect="None" Align="Left" Font="12x7">
20                 201
21             </Number>
22             <Text Effect="None" Align="Left" Font="5x5">
23                 PR. D. JOÃO I
24             </Text>
25         </Page>
26     </Panel>
27 </Destination>
28 </Line>
29 </Group>
30 </GID>
```

Onde as tags presentes no código Xml (3.1) significam:

- ID - Identificador de referência do GID, Grupo, Linha ou Destino
- Name - Título da Linha ou Destino
- Lines - Número de Linhas de LED presente no painel
- Columns - Número de colunas de Led presente no painel
- SBC - Espaço entre Caracteres da página
- SBNAnc - Espaço entre o Número da Linha e os campos de texto da Página
- PT - Tempo de fixação da Página
- Effect - Efeito aplicado no campo
- Align - Alinhamento do texto no campo

- Font - Tamanho da fonte aplicada ao campo

O *software* DMSConsole deve ser capaz de ler um arquivo XML com as informações da estrutura de dado GID, como apresentado em (3.1), e ainda quando necessário criar um arquivo XML semelhante. O DMSConsole deve possibilitar ao usuário criar, através da interface de usuário, um arquivo XML com a estrutura de dados GID, para servir de fonte de dados. Existem varias bibliotecas prontas em C++ para leitura e criação de arquivos XML, porém para tornam as funções de leitura e criação mais eficientes e reduzir a complexabilidade do código, fora desenvolvido uma biblioteca própria nomeada GIDXml.

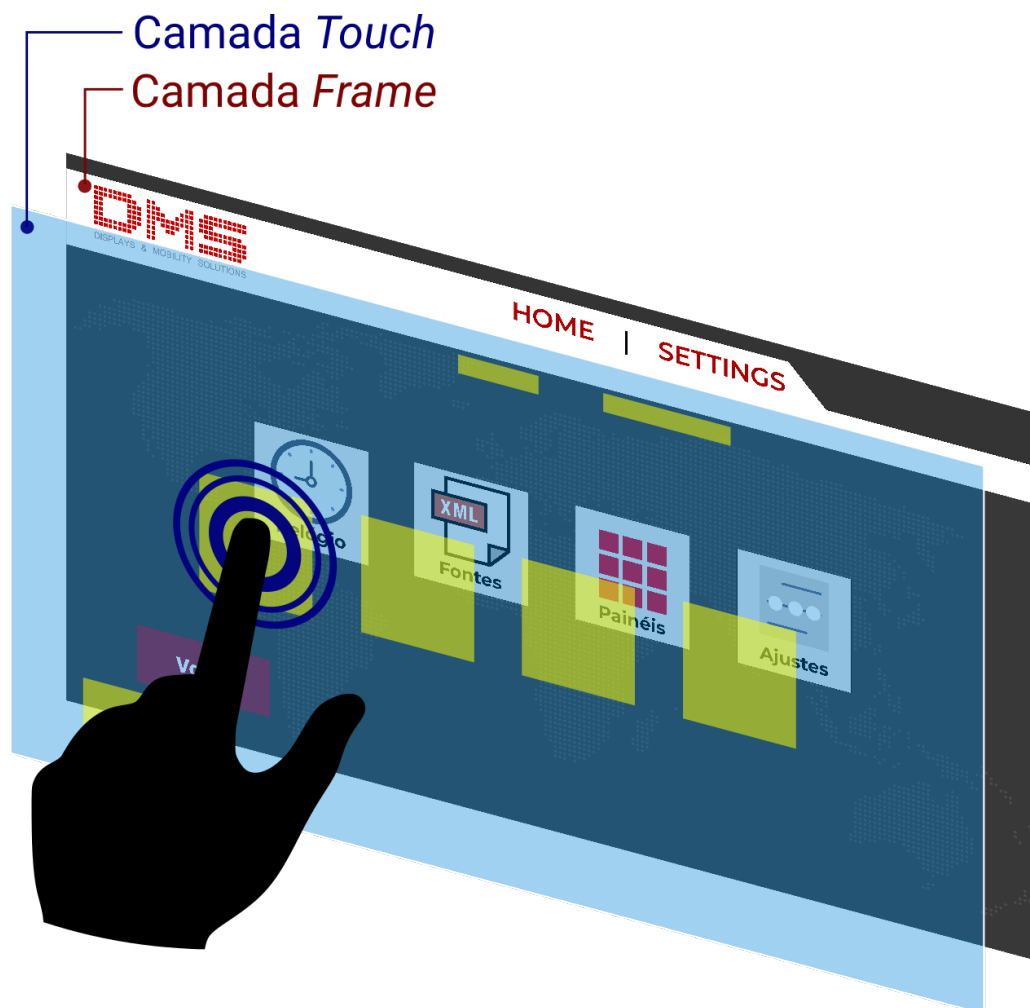
As funções de leitura e criação da biblioteca GIDXml utilizam uma estrutura de dados semelhante a GID, porém com um sistema de alocação dinâmica de memória. A alocação dinâmica permite uma melhor gestão e maior capacidade de uso da memória da aplicação, tornando possível a aplicação ler arquivos XML de maior volume.

## 3.2 GERENCIADOR DE INTERFACE

A interface do usuário é um dos elementos principais deste trabalho, e o principal diferencial deste novo console, em comparação ao modelo SICON II. Esta nova interface faz uso do *touchscreen* como recurso visual e de interação entre console e motorista, de modo a deixar as informações visuais mais legíveis e a navegação pela interface mais simplificada.

O desenvolvimento da interface de usuário utilizando o módulo TOPWAY HKT080ATA-C é baseado em páginas e camadas. Uma página é o conjunto de elementos visuais e recursos de interação com o usuário que se apresentam na tela do *display* de acordo com a navegação pelos recursos da interface. Cada página possui duas camadas; a camada *touch* e a camada *frame*. A dinâmica da interface é feita ao combinar as camadas *frame* e *touch* em diferentes páginas.

A camada *touch* representa a superfície do *display* sensível ao toque. Nesta camada a tela é preenchida com uma página, a qual é dividida em regiões com IDs distintos. Quanto tocadas estas regiões ativam no módulo o envio de mensagens para o dispositivo mestre, afim de informar o ID da região tocada e o ID da página exibida. A camada *frame* representa o ambiente onde são montadas as imagens exibidas na página atual, de modo a tornar visível ao usuário as regiões sensíveis ao toque, como mostrado na Figura (16).



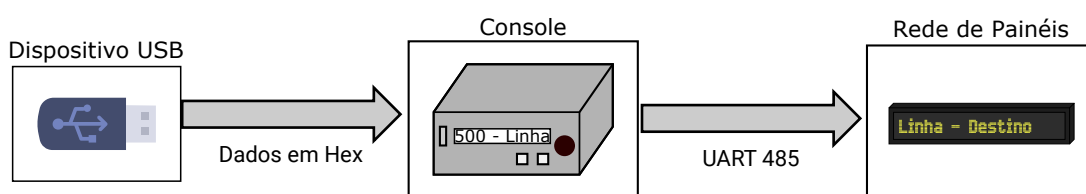
**Figura 16:** Interface no *Display Touchscreen DMS*  
**Fonte:** Autoria Própria

Para cada página da interface há uma camada *touch* e a uma camada *frame* associadas. Quanto maior for o número de páginas necessárias para oferecer as funcionalidade da interface ao usuário, mais extenso será o trabalho de desenvolver a interface. Para tornar a tarefa de desenvolvimento menos penosa, a interface foi pensada de modo a reduzir o número de páginas, e reunir o máximo de ferramentas para o usuário em uma mesma página, sem afetar a visibilidade e o uso das mesmas.

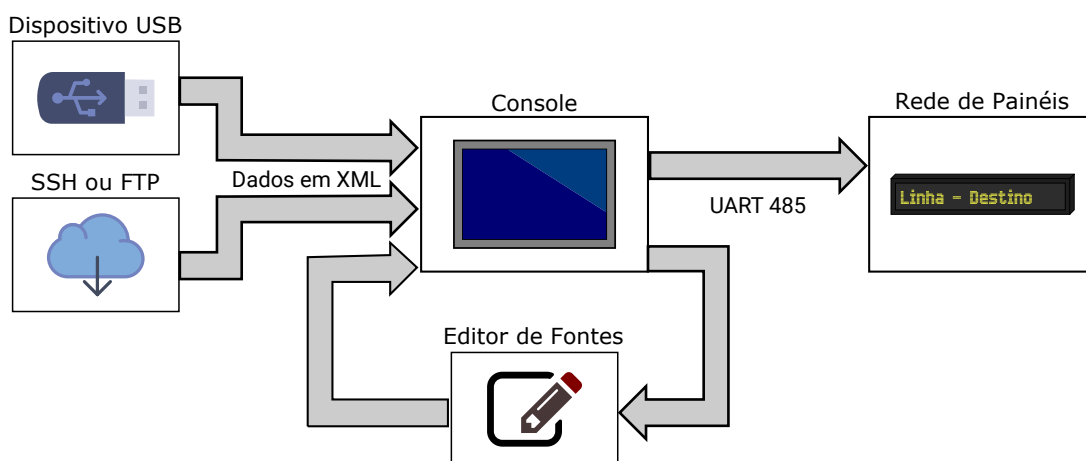
O console de indicadores de destino deve possuir uma interface que facilite ao motorista configurar os painéis e manipular as informações a serem exibidas de forma mais prática o possível. Deve se ter em mente que a usabilidade do console esta intimamente ligada ao tipo de rotina que o autocarro possui. Se o autocarro realiza sempre o mesmo trajeto, em um ambiente urbano pro exemplo, provavelmente as configurações e personalizações das informações fixadas nos painéis raramente mudem. Sendo assim é mais prático utilizar a aplicação GID em um computador, e

apenas exportar o arquivo fonte para um dispositivo USB e carregá-lo no console. Porém quando o trajeto do autocarro varia frequentemente, ou ainda há uma necessidade periódica de alterar informações dos painéis, se torna mais interessante existir uma ferramenta no próprio console que possibilite criar um arquivo fonte.

No antigo modelo de console SICON II não há editor de fontes no console, e conseqüentemente a interface do usuário era mais simples. Já que os dados fonte eram passados via conexão USB, em formato binário, para dentro do console. No novo modelo de console haverá 3 opções de origem das fontes de dados; transferência via USB, transferência via SSH e a criação da fonte direto no console, todos em formato XML (Figura 18). Assim a interface do novo console deve agregar novas funcionalidade, porém sem se tornar demasiadamente complexa.



**Figura 17: Fonte de Dados - SICON II**  
**Fonte: Aatoria Própria**

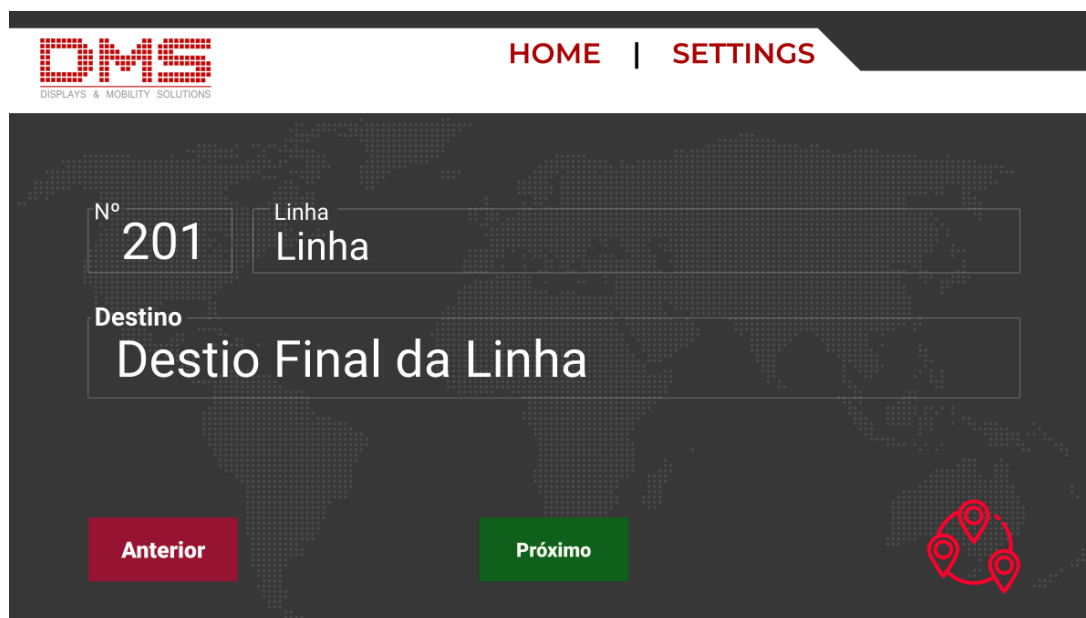


**Figura 18: Fonte de Dados - Novo Console**  
**Fonte: Aatoria Própria**

### 3.2.1 PÁGINA DE SELEÇÃO DE DESTINO

A primeira página da interface a ser construída é a página de Seleção de Destino, Figura 19. Nesta página, dado um arquivo de fonte previamente carregado e uma linha selecionada, o motorista pode selecionar qual é o destino final desta linha.

Assim que o motorista seleciona o destino final, o console busca dentro da estrutura de dados do destino o conjunto de painéis que possuem as dimensões compatíveis com os painéis disponíveis da rede no momento, e envia os dados para os painéis compatíveis.



**Figura 19: Página de Seleção de Destino - DMSConsole**  
**Fonte: Autoria Própria**

### 3.2.2 PÁGINA DE SELEÇÃO DE LINHA

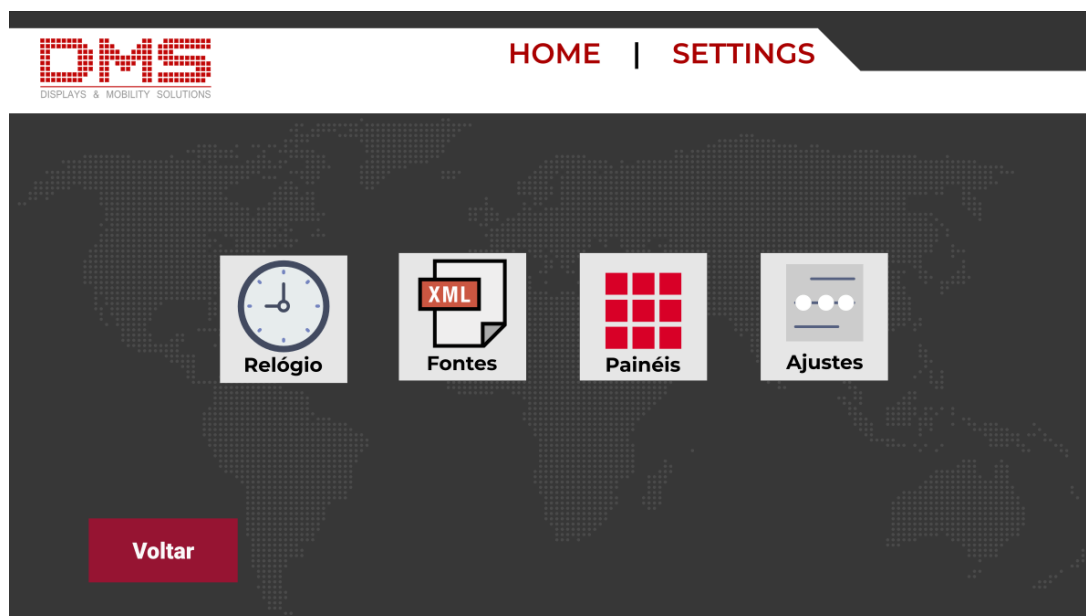
Através do botão de mudança de linha (botão vermelho a direita da tela), presente na página Seleção de Destinos, o motorista tem acesso a página de Seleção de Linha, Figura (20). Nesta página o motorista pode selecionar uma linha para o autocarro dentre as linha presentes no arquivo fonte previamente carregado. O motorista pode utilizar o teclado virtual desta página para buscar a linha de interesse pelo seu número.



**Figura 20: Página de Seleção de Linha - DMSConsole**  
**Fonte: Autoria Própria**

### 3.2.3 PÁGINA DO MENU DE CONFIGURAÇÕES

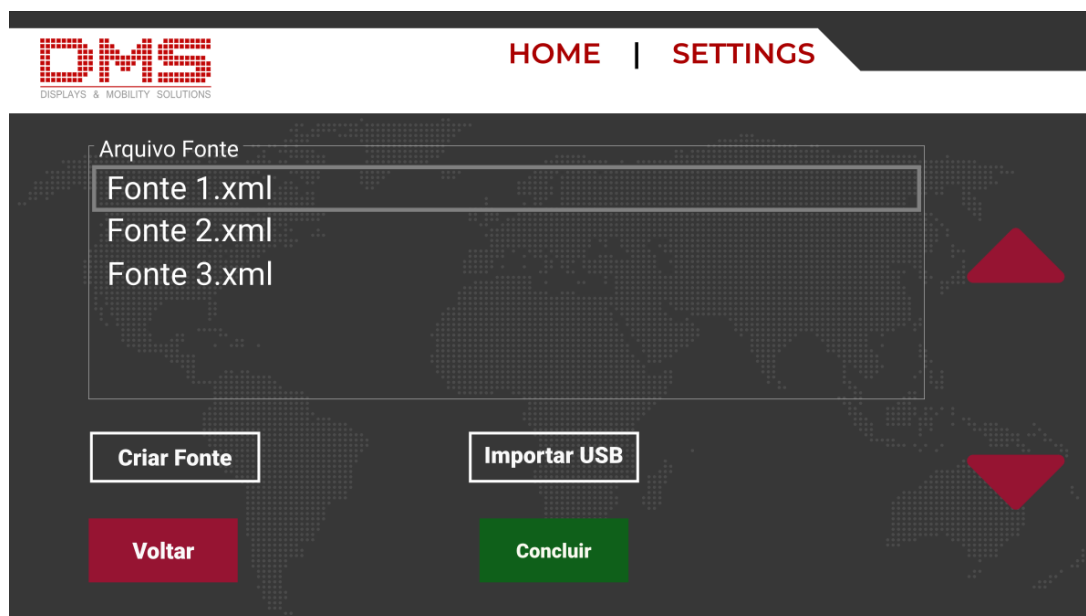
Para acessar as configurações do DMSConsole, o usuário através do botão 'Settings', no topo de qualquer página da interface, pode acessar o Menu de Configurações, Figura 21. Nesta página o usuário tem acesso a página de Seleção de Fonte e as páginas de configuração do RTC (*Real Clock Time*), *Display Touchscreen* e do painéis eletrônicos da rede.



**Figura 21: Página do Menu de Configurações - DMSConsole**  
**Fonte: Autoria Própria**

#### 3.2.4 PÁGINA DE SELEÇÃO DE FONTES

Na página de Seleção de Fonte o usuário tem acesso a uma lista de arquivos fonte em formato Xml que já foram previamente carregados no diretório padrão e que podem ser escolhidos como nova fonte de dados, Figura 22. Para carregar um novo arquivo fonte para este diretório o usuário pode escolher se conectar ao Raspberry Pi via SSH e copiar o arquivo fonte para dentro do diretório 'Data' no diretório de instalação do DMSConsole. Também há a opção de importar diretamente de um dispositivo de armazenamento USB uma nova fonte. Porém ao selecionar um arquivo o DMSConsole verifica a estrutura Xml do arquivo antes de importa-lo, notificando ao usuário se a estrutura é válida ou não. Ainda há a terceira opção, a de criar um arquivo fonte Xml.

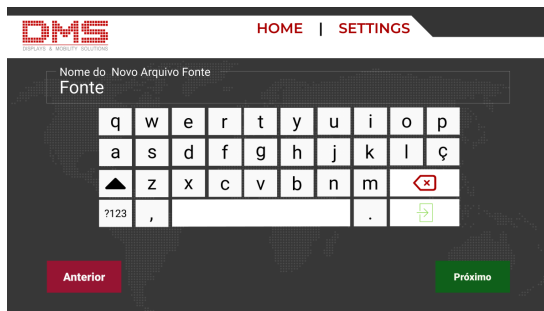


**Figura 22: Página de Seleção de Fonte - DMSConsole**  
**Fonte: Autoria Própria**

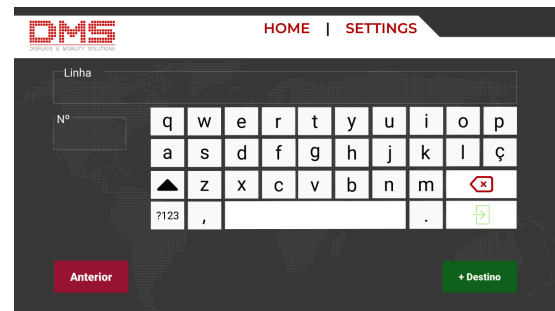
### 3.2.5 PÁGINA DE EDIÇÃO DE FONTES

Ao escolher a opção 'Criar Fonte', dentro da Página de Seleção de Fontes, o usuário é direcionado para editor de fontes do DMSConsole. Onde ele define os parâmetro da Linha e do Destino do autocarro, os parâmetros de dimensões dos painéis, o layout das páginas e o alinhamento das informações visuais nos painéis. E então o DMSConsole gera um arquivo Xml baseados nestas informações, para ser utilizado como fonte. Para facilitar a entrada de dados pelo usuário as páginas do editor de fontes possui um teclado virtual adaptado do padrão *QWERTY*.

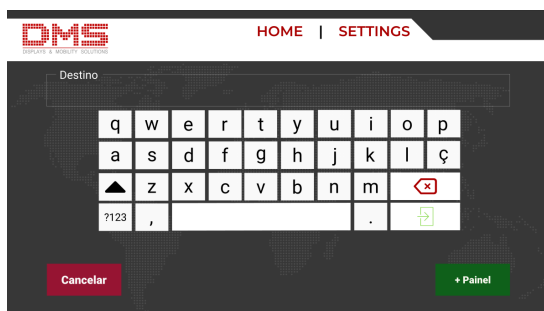
Após inserir o nome do arquivo, o nome e número da linha e o nome do destino, o DMSConsole detecta as configurações dos painéis da rede, e oferece ao usuário a criação de 'Paginas' de acordo com a dimensão de cada painel. Porém se não for detectado nenhum painel da rede, o usuário pode inserir manualmente as dimensões dos painéis. No processo de criação da página o usuário tem a opção de escolher quais informações serão exibidas nos painéis; texto, apenas o número da linha, o texto e o número da linha ou sem nenhuma informação. É possível também definir manualmente a fonte, o alinhamento e o conteúdo do texto exibido. O quadro de Figuras 23 apresenta as etapas do editor de fontes.



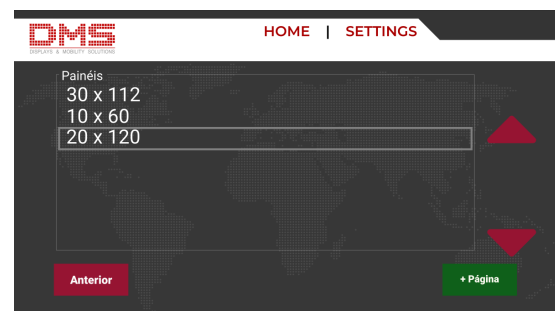
(a) Tela de Nomeação do Arquivo Fonte



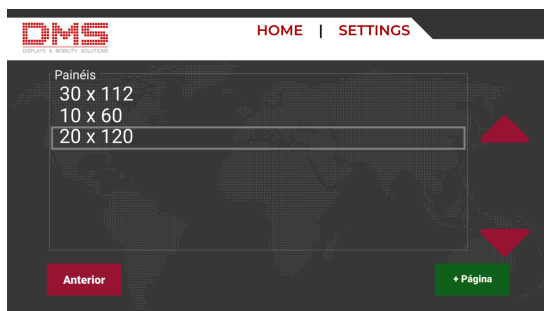
(b) Tela de Criação de Uma Linha de Auto-carro



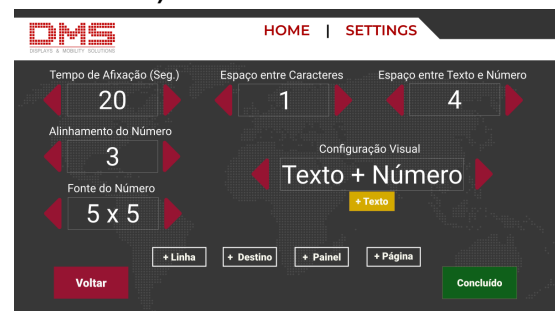
(c) Tela de Criação de Destino



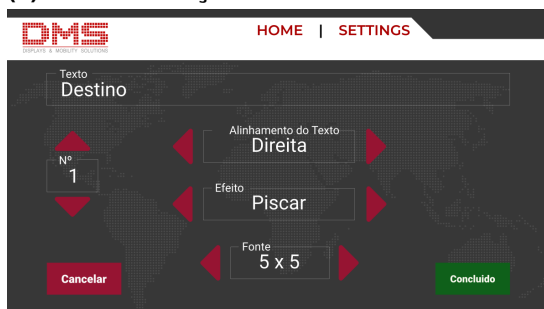
(d) Tela de Criação de Painel (Painéis Não Detectados)



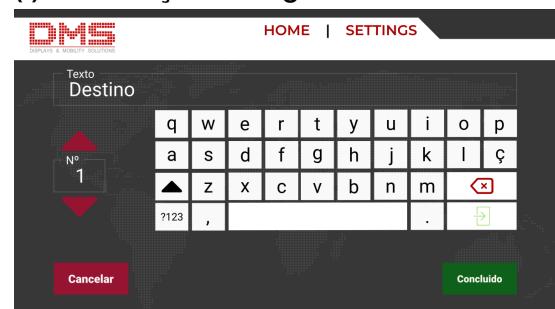
(e) Tela de Criação de Painel



(f) Tela Criação de Página



(g) Tela de Criação de Campo de Texto - Configuração



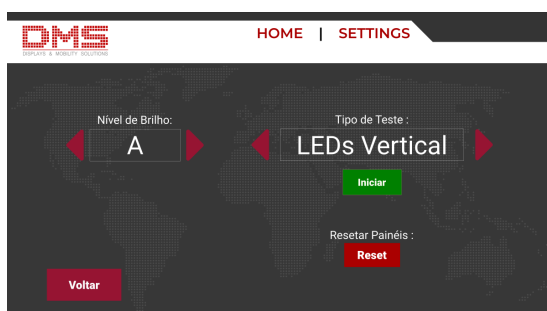
(h) Tela de Criação de Campo de Texto

Figura 23: Editor de Fontes DMSConsole  
Fonte: Autoria Própria

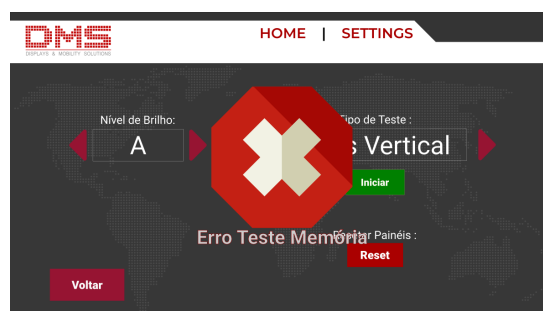
### 3.2.6 PÁGINA DE CONFIGURAÇÃO DOS PAINÉIS

Na Página de Configuração dos Painéis o usuário tem acesso a configuração de nível de brilho dos painéis na rede, a execução de testes de funcionamento dos LEDs e da Memória Flash do equipamento e ainda a realização do *reset* de todos os painéis na rede, Figura (24). O brilho dos painéis é definido em 6 níveis mais o nível automático, onde o brilho do painéis é regulado pelo sensor de luminosidade do equipamento.

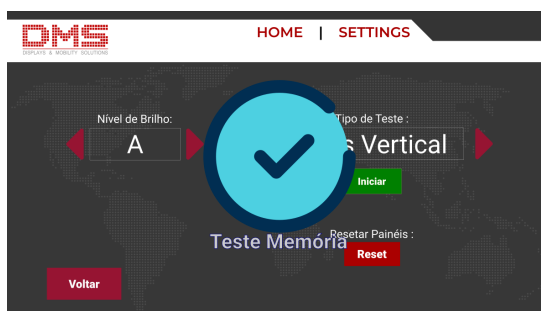
A 3 teste de painéis disponíveis na página de Configuração dos Painéis; o teste de varrimento vertical, varrimento horizontal e teste de memória RAM e FLASH. Caso o teste escolhido seja o teste de memória RAM e Flash o console envia o comando 'MSG Panel Test' para todos os painéis da rede. Se todos os painéis realizarem o teste com sucesso, um ícone aparece na tela informando o sucesso da operação, caso contrário é informado a ocorrência de um erro.



(a) Página de Configuração dos Painéis  
Fonte: Autoria Própria



(b) Erro no Teste de RAM e Flash  
Fonte: Autoria Própria



(c) Sucesso no Teste de RAM e Flash  
Fonte: Autoria Própria

**Figura 24: Página de Configuração dos Painéis - DMSConsole**  
Fonte: Autoria Própria

### 3.2.7 PÁGINA DE CONFIGURAÇÃO DO DISPLAY

Na página de Configurações de *Display* o usuário tem acesso ao *Backlight*, *Screen Saver* e *Buzzer Touch Sound*. O *Backlight*, ou retro-iluminação, é a intensidade da fonte de luminosa de fundo do *Display*. Uma intensidade elevada de *Backlight* torna o conteúdo mostrado no *Display* mais visível ao usuário em um ambiente bastante iluminado. Porém em um ambiente mais escuro um nível alto de *Backlight* pode incomodar o usuário é causar mais rapidamente cansaço visual. Na página de configuração do *Display* é possível selecionar o nível do *Backlight*, e ainda definir para dois períodos do dia a intensidade de desejada.

O *Screen Saver* é o recurso que reduz o *Backlight* para zero quando o *Display* esta ocioso. Na página de configuração do *Display* é possível definir um valor de até 900 segundos para que o tempo máximo de ociosidade para que o *Display* não entre em *Screen Saver*. Quando o *Screen Saver* é definido como 0, o *Backlight* nunca é reduzido .

Por último existe o *Buzzer Touch Sound*, este é um recurso sonoro que informa ao usuário quando uma tecla *touch* foi pressionada. O tempo de emissão sonora pode ser configurado nesta página, sendo a unidade definida em 10 milésimos de segundos.

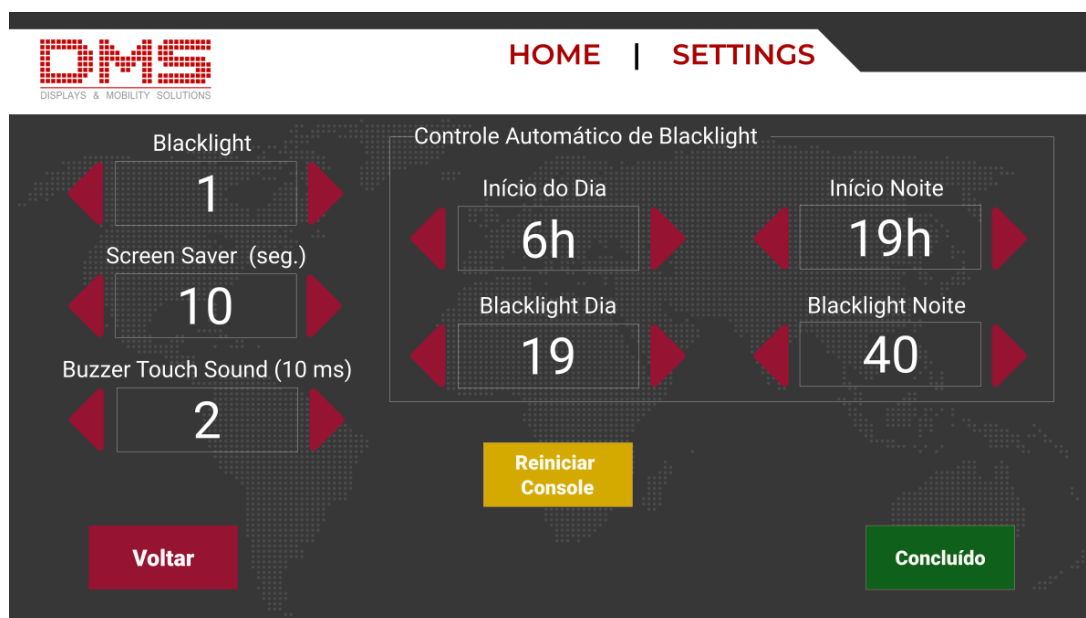


Figura 25: Página de Configuração do Console - DMSConsole  
Fonte: Autoria Própria

### 3.2.8 GERENCIADOR PAINÉIS

Para que a aplicação DMSConsole seja capaz de gerenciar a rede de painéis eletrônicos conectados ao console, e ainda atender as requisições feitas através da interface de usuário, é desenvolvida a biblioteca DMSDisplay. Esta biblioteca contém as funções capazes de efetuar o reconhecimento e configurações dos dispositivos da rede de indicadores de destino. Para tal esta biblioteca se utiliza de uma interface de comunicação serial emulada e dois *timers* do sistema Linux. Estes *timers* são úteis para realizar algumas rotinas de configurações e manter o procedimento cíclico de atualização dos painéis.

O gerenciamento dos painéis é feito através do envio e recepção de mensagens usando o protocolo de comunicação LDESIGN. O console é um dispositivo *Master* que realiza as requisições aos painéis, com mensagens console-painéis, e os painéis respondem com mensagens painel-console. O DMSConsole deve enviar e receber tais mensagens, mediante a requisição do usuário e/ou as necessidades que as rotinas de configurações exigem. Dentro do LDESIGN existe um conjunto de tipos de mensagens, sendo que para cada tipo é desenvolvido dentro da biblioteca DMSDisplay uma função C++ para executar a transmissão ou recepção dos dados. As subseções a seguir apresentam os tipos de mensagem deste protocolo, definindo o valor para os campos 'Tipo' e 'Dados', e especificando a funcionalidade de cada mensagem para o DMSConsole.

#### 3.2.8.1 MSG NETWORK CONFIG

Esta mensagem realiza o reconhecimento dos dispositivos conectados a rede de painéis eletrônicos. Esta mensagem é transmitida pelo console em *broadcast* usando o endereço 0xFF a todos os painéis, os quais respondem a requisição com uma mensagem 'Painel ID'(3.2.8.7), em sequência de acordo com o seu ID. O byte 'Tipo' desta mensagem é definido como 0x22, 'N' é igual a 2, e não existem bytes a serem enviados no campo 'Dados'. Ao final de cada Painel ID ainda é enviado um byte 'ACK' ou 'NACK' de reconhecimento.

Quando o DMSConsole é iniciado ele realiza uma rotina que envia uma mensagem *MSG Network Config* a todos os painéis via *broadcast*, de modo a detectar o número e as configurações dos painéis ligados a rede de indicadores de destino. Se nenhum painel for detectado se faz necessário informar ao usuário tal informação, caso contrário o usuário só será informado do número ou configurações dos painéis

quando for necessário criar um arquivo de fonte de dados pelo editor do DMSConsole.

### 3.2.8.2 MSG DEFINE BRIGHTNESS

Através desta mensagem é possível configurar o nível de brilho de cada painel. Esta mensagem é enviada pelo console e pode ser endereçada para um painel específico, ou enviada por *broadcast* via endereço 0xFF. O byte 'Tipo' desta mensagem é 0x26, e 'N' é igual 2. O comando 'Dados' é definido como 1 byte em ASCII que representam a intensidade do brilho a ser utilizado no painel, sendo definido:

- '1' Nível 1 de Brilho
- '2' Nível 2 de Brilho
- '3' Nível 3 de Brilho
- '4' Nível 4 de Brilho
- '5' Nível 5 de Brilho
- '6' Nível 5 de Brilho
- 'A' Nível Automático

Em resposta a esta mensagem o painel deve enviar uma mensagem 'ACK' em caso reconhecer e validar a mensagem, ou caso contrário deve ser enviado uma mensagem 'NACK'. O DMSConsole deve enviar esta mensagem apenas quando o usuário acender a Página de Configuração dos Painéis e selecionar um nível de brilho diferente do atual.

### 3.2.8.3 MSG RESET

Esta mensagem destina-se a realização de limpeza na memória dos painéis eletrônicos, colocando estes no estado de arranque. O byte tipo desta mensagem é definido como 0x1A, o byte 'N' é 1, não existe bytes a serem enviados no campo 'Dados'. Esta mensagem é enviada pelo console, sendo destinada a todos os painéis, via transmissão *broadcast*. Após o envio desta mensagem o painel deve responder com uma mensagem 'ACK' se este reconhecer a mensagem, caso contrário ele deve mandar uma mensagem 'NACK'. Esta mensagem só é enviado pelo DMSConsole quando o usuário acende a Página de Configuração dos Painéis é seleciona a opção 'Reset'.

## 3.2.8.4 MSG INFO

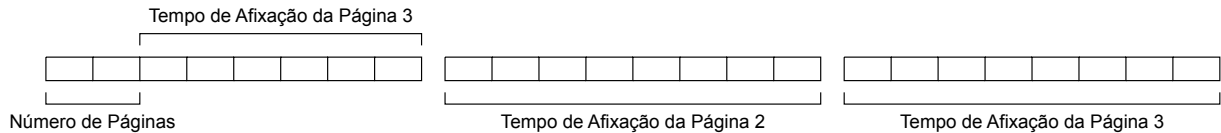
Esta mensagem é enviada pelo console e contém dados que descrevem as informações a serem afixadas em um painel específico. Esta mensagem é enviada individualmente para cada painel. O byte 'Tipo' desta mensagem é definido como 0x21, e o Byte 'N' é variável de acordo com o número total de bytes presentes nessa mensagem. O campos 'Dados' segue uma estrutura geral descrita abaixo:

Ordem	Comando	Descrição	Nº Bytes
1º	Index	Contador de Referência	1
2º	Page General Info	Informações gerais sobre as páginas a seguir	3
3º	Stream ID	Define se a mensagem é BITMAP ou ASCII	1 1
4º	Page Info 1	Descrição da Pagina 1	N
5º	Page Info 2	Descrição da Pagina 2 (Opcional)	N
6º	Page Info 3	Descrição da Pagina 3 (Opcional)	N
7º	<END>	<i>End of Message</i> - 23h	1

**Tabela 3: Formato Geral - MSG Info**  
**Fonte: DMS (2004)**

Cada painel possui um contador *TimeOut* que cronometra o tempo decorrido desde a última mensagem MSG INFO enviada. Quando este contador atingi o valor de 2 minutos o painel apaga as informações fixadas. Assim, no DMSConsole deve haver uma rotina que, no máximo a cada 2 minutos, reenvia automaticamente mensagens do tipo MSG INFO a todos os painéis da rede, mesmo que estas mensagens sejam iguais as últimas mensagens enviadas. Para que o *firmware* dos painéis não precise apagar e reescrever os dados fixados toda a vez em que mensagens tipo MSG INFO exatamente iguais são enviadas, com o intuito de evitar que o contador *TimeOut* estoure, o primeiro Byte da mensagem, o 'Index', somente é incrementado quando a mensagem a ser enviada é diferente da anterior.

O comando 'Page Geral Info' é o conjunto de 3 bytes que descrevem as informações a cerca das páginas a serem fixadas no painel de destino, a Figura (26) apresenta a descrição deste bytes na ordem MSB (*Most Significant Bit*).



**Figura 26: MSG Info - Page General Info**  
**Fonte: DMS (2004)**

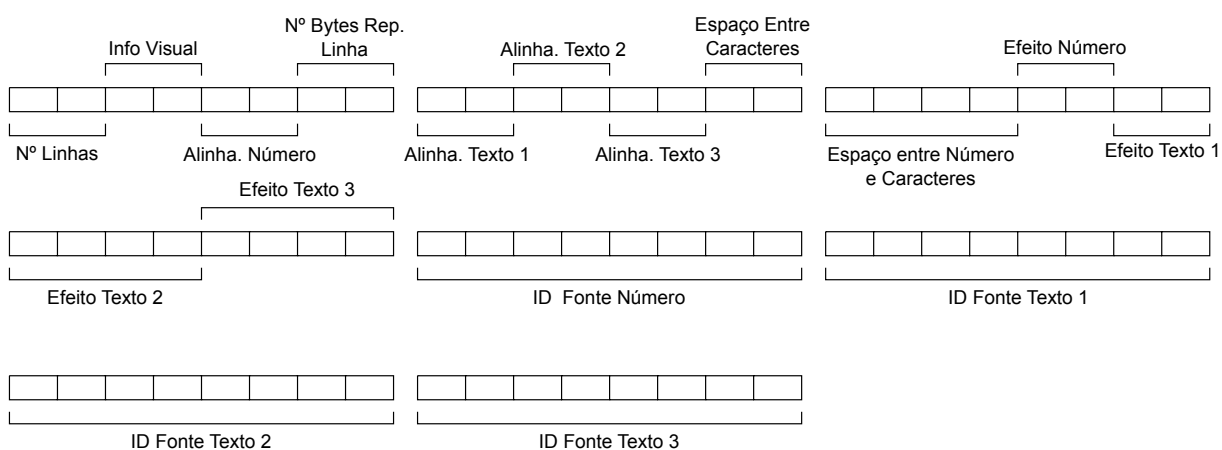
O comando 'STREAM ID' tem dois formatos diferentes, quando a mensagem é no sentido painel-console 'STREAM ID' é composto com 1 byte ASCII com o caractere 'A' quando a informação é no formato ASCII, ou 'B' quando é no formato BITMAP. Já se a mensagem é no sentido console-painel o comando 'STREAM ID' é composto por 5 bytes, se estes bytes forem preenchidos corretamente com a sequência de caracteres 'ASCII' a mensagem é classificada como ASCII, ou caso contrário ela é classificada como BITMAP.

O comando 'Page Info' descreve as informações a serem visualizadas nos painéis, sendo estas informações divididas por páginas, as quais podem possuir o número da linha e até 3 campos de texto. Em um painel é permitido de 1 a 3 páginas, sendo obrigatório haver ao menos uma página. A estrutura do comando 'Page Info' é descrita na Tabela 4.

Ordem	Comando	Descrição	Nº Bytes
1º	Page Header	Cabeçalho de definição das informações sobre a página	3
2º	Bus Line	Conjunto de Bytes em ASCII que representam o número da Linha	N
3º	Info Line 1	Conjunto de Bytes em ASCII que representam o texto 1 a ser afixado (Opcional)	N
4º	<CR ><LF>	Separadores de blocos Page Info - 0x0D 0x0A (Opcional)	2
5º	Info Line 2	Conjunto de Bytes em ASCII que representam o texto 2 a ser afixado (Opcional)	N
6º	<CR ><LF>	Separadores de blocos Page Info - 0x0D 0x0A (Opcional)	2
7º	Info Line 3	Conjunto de Bytes em ASCII que representam o texto 3 a ser afixado (Opcional)	N

**Tabela 4: Formato Page Info - MSG Info**  
**Fonte: DMS (2004)**

O 'Page Header' é o cabeçalho que descreve cada página a ser transmitida, sendo que este comando possui apenas 3 bytes, a serem preenchidos na seguinte forma:



**Figura 27: MSG Info - Page Info Header**  
**Fonte: DMS (2004)**

Onde:

- **Nº Linhas** : Número de campos de texto na página, de 0 a 3 páginas.
- **Info Visual** : Formato das informações a serem exibidas, sendo as opções:
  - 00 - Apenas o número da linha
  - 01 - Apenas os campos de Texto
  - 10 - O número da linha e os campos de texto
  - 11 - Nenhuma informação
- **Alinha. Número / Alinha. Texto 1,2 e 3** : Alinhamento do número da linha e dos campos de texto, sendo as opções para cada elemento:
  - 00 - Esquerda
  - 01 - Centro
  - 10 - Direita
  - 11 - Não identificado
- **Efeito Número / Efeito Texto 1,2 e 3** : Efeito a ser aplicado no número da linha e nos campos de texto, sendo as opções para cada elemento:
  - 000 - Sem efeito
  - 001 - Piscar
  - 010 - Rodar
  - 011 - Não definido
- **ID Fonte Número / ID Fonte Texto 1, 2 e 3** : Tamanho da fonte a ser aplicada no número da linha e nos campos de texto. Estas fontes são carregadas no Firmware dos painéis e podem mudar de acordo com a versão do painel. Na maioria dos modelos de painéis eletrônicos da DMS as opções de fonte são:
  - 0x00 - 5x5
  - 0x01 - 7x5
  - 0x02 - 8x6
  - 0x03 - 11x7
  - 0x04 - 12x7
  - 0x05 - 13x9
  - 0x06 - 14x9

0x07 - 15x9

0x08 - 16x9

0x09 - 17x11

0x10 - 19x11

O comando 'Page Info' deve ser seguido pelo comando 'Bus Line', o qual nada mais é do que o conjunto de caracteres ASCII que representam o número da linha. Deve-se observar que o número de bytes 'Bus Line' deve ser o mesmo já definido em 'Page Header'. Após o comando 'Bus Line' se existir o campo de texto 1 é necessário enviar o conjunto de caracteres ASCII que representam a mensagem do texto 1, observando que o limite de 29 caracteres. Se existir um texto 2 é necessário enviar o comando <CR ><LF> para separar os blocos, o mesmo deve ocorrer com o texto 3. Após o envio do MSG Info o painel alvo da mensagem deve enviar um comando 'ACK', se este reconhecer a mensagem e a validar, se não ele envia um comando 'NACK'.

#### 3.2.8.5 MSG PANEL TEST

Esta mensagem destina-se a realização de testes básicos nos painéis. Esta mensagem pode ser endereçada a um painel ou enviada em *broadcast* via endereço 0xFF a todos os painéis da rede. O comando 'Tipo' desta mensagem é definido como 0x40, e 'N' é igual a 2. O comando 'Dados' desta mensagem define qual tipo de teste será realizado, sendo da seguinte forma:

- 0x01 - Teste de RAM e FLASH
- 0x02 - Teste de varrimento vertical de LEDs
- 0x03 - Teste de varrimento horizontal de LEDs

O DMSConsole envia este tipo de mensagem quando o usuário acessa a Página de Configuração dos Painéis e seleciona uma opção de teste. Se o teste escolhido for o teste de varrimento horizontal ou vertical de LEDs, todos os painéis da rede ao receberem esta mensagem executam a rotina de apagar todos os LEDs e acender apenas uma linha, no caso horizontal, ou uma coluna, no caso vertical, por vez varrendo todos os LEDs. Ao final todos os painéis devem manter todos os seus LEDs acesos com brilho nível 2 até receberem novas mensagens.

Já no caso de um teste de memória RAM e Flash, todos os painéis da rede ao receberem esta mensagem passam a checar suas memórias, e cada uma envia sequencialmente uma mensagem ACK caso o teste ocorra com sucesso, ou NACK caso tenham encontrado algum problema durante o teste.

#### 3.2.8.6 ACK E NACK

Esta mensagem é enviada por um painel em resposta a requisição realizada pelo console. A mensagem ACK ou *acknowledge* informa ao console que a mensagem recebida foi reconhecida e validada. Já a mensagem NACK ou *not acknowledge* informa ao console que a mensagem recebida não foi reconhecida. A mensagem 'ACK' tem o campo 'Tipo' definido por 0x86 e 'N' definido por 2. Já a mensagem 'NACK' tem 'Tipo' igual a 0x95 e também 'N' igual a 2. Ambas as mensagens não possuem informação 'Dados'.

#### 3.2.8.7 PANEL ID

Esta mensagem é enviada por uma painel em resposta a uma mensagem MSG Network Config 3.2.8.1. Através desta mensagem o console identificar as configurações físicas dos painéis conectados a rede. O comando 'Tipo' desta mensagem é definido como 0x61 e 'N' é igual a 5. O comando 'Dados' desta mensagem é constituída da seguinte forma:

Ordem	Comando	Descrição	Nº bits
1º	Endereço	Endereço do painel na rede	1
2º	Linhas	Número de linhas físicas de LED do painel	1
3º	Colunas	Número de colunas físicas de LED do painel	1
4º	Aux	Define se o painel suporta destinos alternativos	1

**Tabela 5: Formato Panel ID**

**Fonte: DMS (2004)**

#### 3.2.9 BIT BANG PARA COMUNICAÇÃO UART-RS485

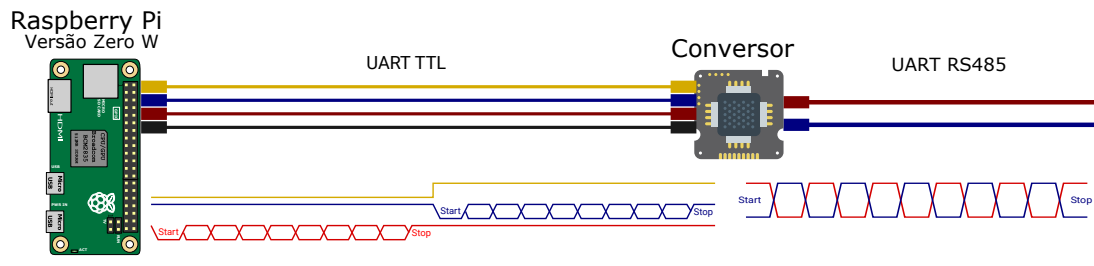
O protocolo LDESIGN é estabelecido em uma interface UART-RS485 com um *Baud Rate* de 9600bps, com 8 bits de dados e 1 *Stop Bit*. Como apresentado

anteriormente, o *Raspberry Pi Zero W* possui apenas uma interface de comunicação UART, o qual já é utilizada neste projeto para a comunicação em UART-RS232 com o *display touchscreen*. Para solucionar este problema é utilizado um recurso de emulação de interface de comunicação por *software* chamado *Bit Bang*.

Utilizando os recursos de leitura e escrita de dados nas portas digitais disponíveis da GPIO, e obedecendo a estrutura do protocolo de comunicação é possível emular, via *software*, o comportamento de uma interface UART, adicionando mais um recurso de comunicação ao *Raspberry Pi Zero W*. Porém um dos obstáculos a ser enfrentado nesta tarefa é a velocidade leitura e processamento dos dados recebidos e transmitidos por essa interface emulada, já que o menor atraso acarreta facilmente erros na transmissão e/ou recepção. Em um sistema que possui um *Hardware* exclusivo para comunicação, como um modem UART, a amostragem dos sinais de entrada, a modulação dos sinais de saída, os níveis de tensão deste sinais, a sincronização e o armazenamento temporário são controladas independentemente da execução do *software* pelo processador. Já no *Bit Bang* todos estes parâmetros devem ser controlados pelo *software*.

Já existem bibliotecas em C++ que implementam o *Bit Bang* com um custo computacional bastante reduzido, mesmo assim o limite da velocidade de comunicação (*Baud Rate*) ainda é limitado e depende da velocidade de processamento do dispositivo embarcado. O *Raspberry Pi Zero W* possui um processador de 1GHz *single core* que prove uma boa velocidade de processamento, e conseqüentemente torna possível emular interfaces de comunicação com um *baud rate* considerável. Nesta aplicação é utilizado a biblioteca *PiGPIO*, disponível em [pigpio \(2018\)](#), o qual oferece uma variedade de funções em C para comunicação serial via *bit bang*.

Os pinos da GPIO do *Raspberry Pi Zero W* possuem um nível de tensão de 0 a 3,3V. Conseqüentemente qualquer interface de comunicação estabelecido estará neste limite de tensão. Já que o UART-485 opera como tensões diferenciais em uma faixa além da oferecida pelo *Raspberry*, fazendo necessário utilizar um conversor UART TLL / UART-485. Neste projeto o conversor utilizado é o CI SN65HVD308 - TI (TEXAS, 2012a), o qual apresenta no lado TLL 4 pinos; porta de envio, recepção, habilitação de envio e aterramento. No lado 485 há apenas duas portas diferenciais A e B. A Figura 28 apresenta o esquemático da conexão entre este conversor e o *Raspberry*.



**Figura 28: Conversor UART TTL - UART 485**  
Fonte: DMS (2004)

Assim através da interface de comunicação emulada via *bit bang*, utilizando a porta 20 da GPIO para a habilitação de envio, a porta 23 para envio da mensagem e 24 para a recepção, a comunicação entre console e painéis é feita no padrão LDESIGN.



## 4 MONTAGEM E TESTE DO PROTÓTIPO

Neste etapa é desenvolvido o *hardware* para o novo console, o qual deve suportar o *software* desenvolvido no capítulo (3). O novo console de indicadores de destino da DMS tem a necessidade de ser alocado dentro do painel dos autocarros, em uma disposição muito parecida com a que o modelo SICON II ocupava. Nesta disposição o novo console deve estar montado dentro de uma caixa suporte com dimensões precisas, e com os conectores de comunicação e alimentação alinhados. Como o projeto da caixa suporte não esta dentro do escopo deste trabalho, o departamento de desenvolvimento da DMS disponibilizou uma caixa suporte de plástico para o módulo *Display Touchscreen* que atende a todas os requisitos de dimensões e de localização dos conectores.

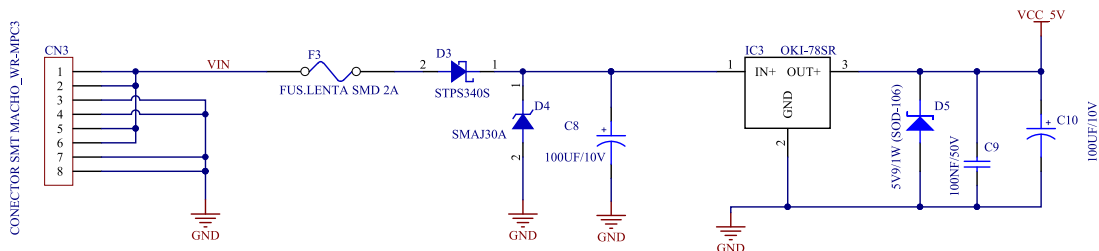
### 4.1 DESENVOLVIMENTO DA PLACA DE INTEGRAÇÃO

Para montar o novo console dentro do suporte em um produto final é necessário projetar uma placa de circuito impresso que faça a integração entre o Raspberry Pi Zero W, os conversores UART (TTL/ RS232) e UART (TTL/ RS485) e uma fonte de tensão de 5V. Um autocarro tipicamente possui tomadas de tensão na faixa dos 24V, na região do painel do motorista. Porém deve se ter cuidado com o fato de que esta tensão é provida pela bateria do autocarro, que por sua vez é alimentada pelo alternador.

Segundo DMS (2004), devido a dinâmica de carga e descarga da bateria alimentada pelo alternador do autocarro, é comum as tomadas de 24V apresentarem variações de tensão de até  $\pm 6V$ . Os conversores UART e o Raspberry devem ser alimentados a uma tensão de 5V, sendo estes dispositivos sensíveis a grandes variações de tensão. A fonte de tensão empregada deve ser um conversor de tensão capaz de absorver as grandes variações, sendo de preferência um conversor CC-CC chaveado, para obter um alto rendimento.

A DMS utiliza o circuito integrado OKI-78SR5, que é um regulador de tensão CC-CC chaveado não isolado, em seus projetos de fontes de alimentação para sistemas embarcados. Este CI pode operar com 7 á 36 V de entrada, mantendo a tensão

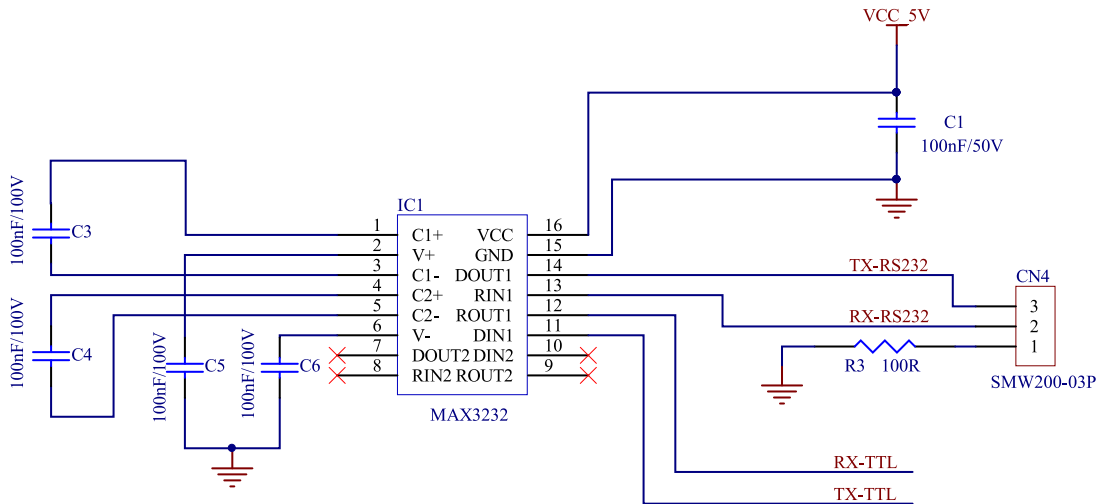
de saída em 5V com uma oscilação de  $\pm 1V$ , logo atendendo a necessidade deste projeto (MURATA, 2010). A Figura (29) apresenta o esquemático do regulador de tensão a ser usado na placa de integração. Este esquemático foi feito no *software Altium*<sup>®</sup>, cedido pela DMS para a realização deste trabalho.



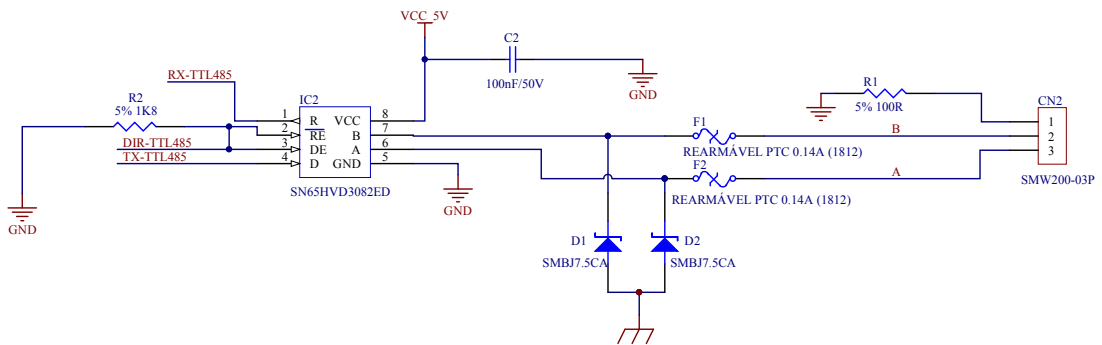
**Figura 29: Esquemático do Conversor CC-CC**  
**Fonte: Autoria Própria**

No esquemático da Figura (29) há um conector de 8 pinos para a tensão de entrada do regulador de tensão, mesmo havendo a necessidade de apenas dois conectores. Isso acontece por uma questão de padronização e disponibilidade de conectores da DMS. Pois esta placa de circuito impresso será montada pelo departamento de produção da DMS, e portanto esta sujeita a limitação de disponibilidade de componentes.

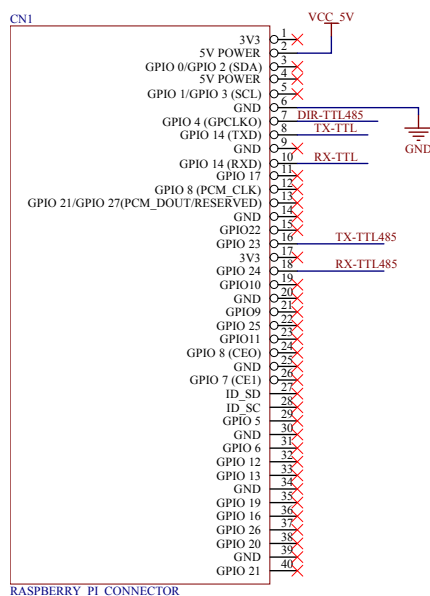
Na mesma folha de projeto do regulador de tensão é feito o esquemático do conversor UART-RS232 e do conversor UART-RS485. Tais esquemáticos obedecem aos requisitos de projeto apresentados nas folhas de informações técnicas dos respectivos componentes; Texas (2012b) e Texas (2012a). As Figuras (30), (31) e (32) apresentam respectivamente os esquemáticos do conversor UART RS232, conversor UART 485 e do conector para Raspberry Pi. O conector presente no esquemático da Figura (32) é um conector fêmea 20x2 pinos, adequado para o encaixe sobre os pinos da GPIO do Raspberry Pi Zero W com pinos macho.



**Figura 30: Esquemático do Conversor UART TTL / UART RS232**  
**Fonte: Autoria Própria**

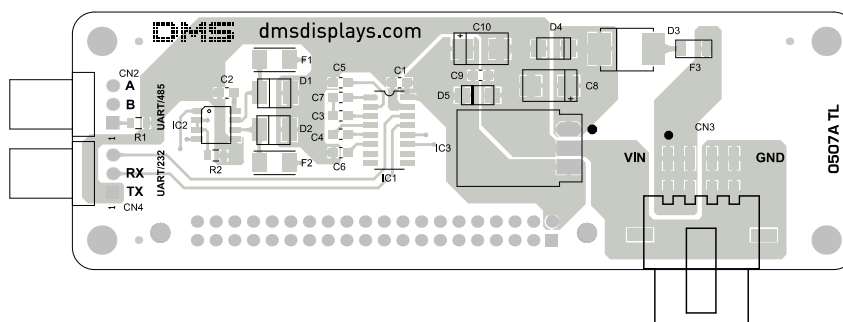


**Figura 31: Esquemático do Conversor UART TTL / UART RS485**  
**Fonte: Autoria Própria**

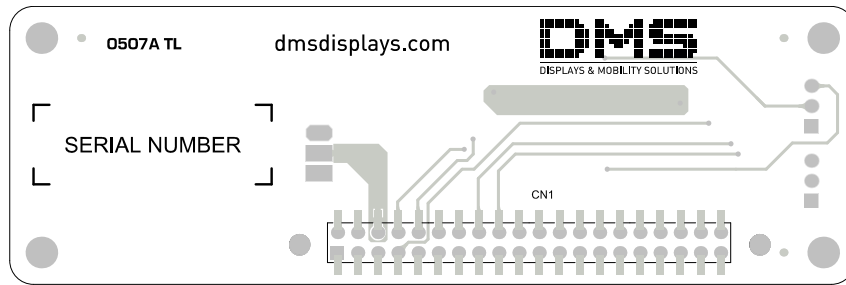


**Figura 32: Esquemático Raspberry Pi Zero W**  
**Fonte: Autoria Própria**

Após montar o esquemático elétrico da placa de integração, é feito o roteamento e o *layout* de montagem da placa. Os componentes utilizados nesta placa são em sua maioria SMD (*Surface Mount Device*), o que reduziu o espaço de placa utilizado e possibilitou o uso de conectores de saída maiores e mais resistentes. As Figuras (33) e (34) apresentam os *layouts* de montagem da placa de integração, mostrando os roteamentos, a serigrafia e os componentes.

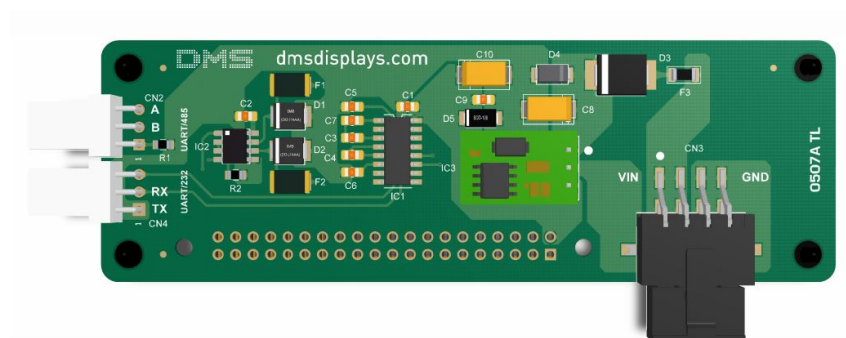


**Figura 33: Layout de Montagem DP0507-0A0 - Superior**  
**Fonte: Autoria Própria**

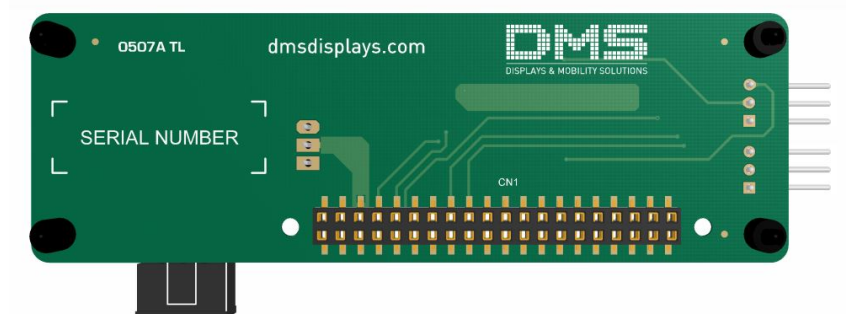


**Figura 34: Layout de Montagem DP0507-0A0 - Inferior**  
Fonte: Autoria Própria

Para facilitar a visualização final as Figuras (35) e (36) apresentam o modelo 3D da placa de integração.



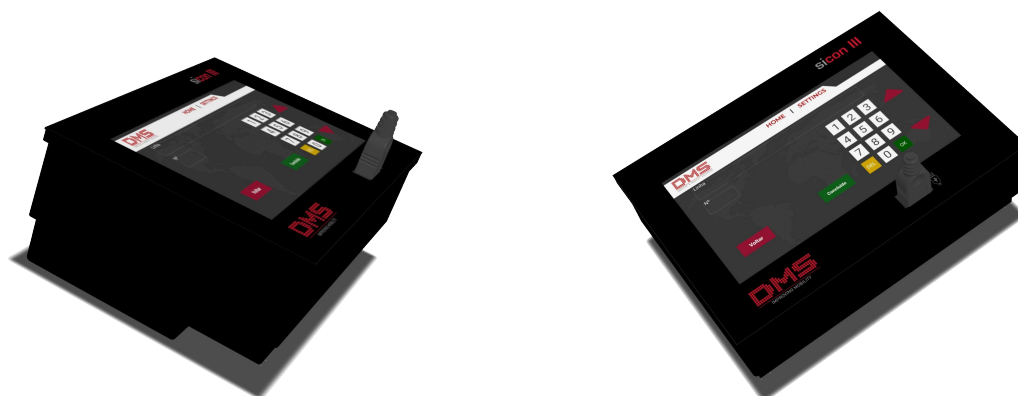
**Figura 35: Modelo 3D DP0507-0A0 - Inferior**  
Fonte: Autoria Própria



**Figura 36: Modelo 3D DP0507-0A0 - Inferior**  
Fonte: Autoria Própria

Para efeito de visualização final do novo console, é feito também um modelo

3D do suporte de plástico com o módulo do *display* instalado. As Figuras (37a) e (37b) apresentam este modelo 3D. Por questões de padronização, o departamento de desenvolvimento da DMS determinou que o novo modelo de console de indicadores de destino, desenvolvido neste trabalho, será denominado SICON III. Dando assim seguimento a linha de consoles SICON II, o qual é o projeto base deste trabalho.



(a) Vista A

(b) Vista B

**Figura 37: Modelo 3D Console Sicon III**

**Fonte: Aatoria Própria**

Ao final, após a fabricação da placa de integração, todos os componentes do novo modelo de console de indicação de destino são montados dentro da caixa suporte. E assim o primeiro protótipo é montado, se semelhando em muito com o modelo 3D apresentado anteriormente, como pode ser visto nas Figuras (38) e (39).



**Figura 38: Protótipo SICON III - Lateral Esquerda**

**Fonte: Autoria Própria**



**Figura 39: Protótipo SICON III - Lateral Direita**

**Fonte: Autoria Própria**

## 4.2 REALIZAÇÃO DOS TESTES

Após a montagem do protótipo é o momento da realização dos testes de interação usuário-console e console-painéis. Por uma questão econômica, a DMS não mantém em estoque nenhum painel eletrônico de indicadores de destino. Porém oportunamente a DMS disponibilizou por um curto período de 3 dias, um painel eletrônico que estava no departamento para manutenções. É por meio deste painel que é realizado os testes de reconhecimento de painéis, configuração no nível do brilho dos painéis e envio de mensagem para afixação de informações.

O teste de reconhecimento de painéis na rede se dá pelo envio em *Broadcast* de uma mensagem *MSG NETWORK CONFIG*, e a conseqüentemente a resposta *Panel ID* sequencial dos painéis. A aplicação *DMSConsole* possui uma rotina de inicialização que obrigatoriamente realiza o reconhecimento dos painéis, armazenando os dados oriundos das respostas *Panel ID*. Se nenhum painel é reconhecido durante a inicialização da aplicação, o *DMSConsole* exibe no *Display* um aviso ao usuário.

O teste de brilho dos painéis é feito pelo envio de uma mensagem *MSG DEFINE BRIGHTNESS* em *broadcast* aos painéis, os quais devem alterar o nível de brilho de acordo com o nível desejado. O *DMSConsole* realiza este tipo de configuração mediante a requisição do usuário na Página de Configuração dos Painéis.

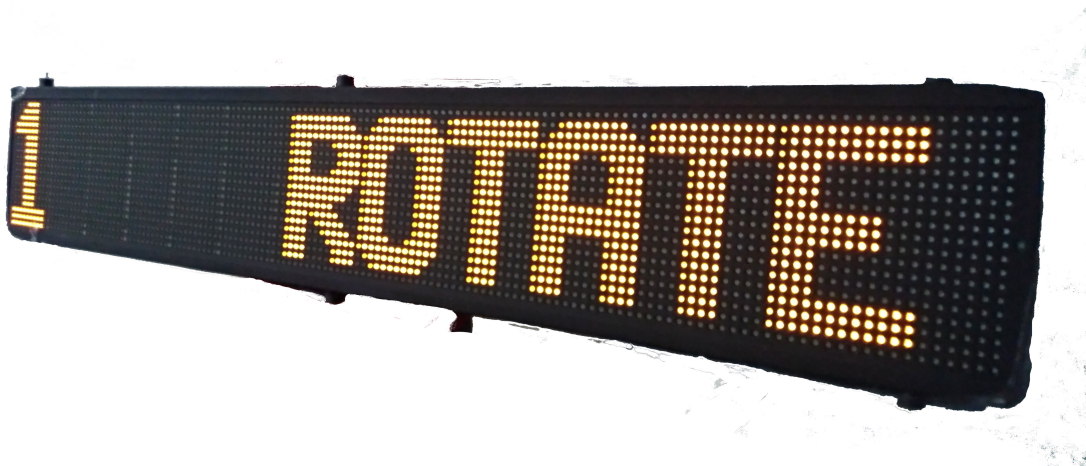
O teste de reconhecimento e brilho são realizados praticamente simultaneamente, já que para configurar o brilho dos painéis é necessário que o *DMSConsole* detecte ao menos um painel na rede, caso contrário ele exibe um sinal de erro ao usuário. A configuração do brilho dos painéis é realizado com sucesso, e o painel utilizado apresenta o nível de brilho desejado, passando assim neste teste.

Em seguida é realizado o teste envio de mensagem para afixação, o qual é feito através do envio de uma mensagem *MSG INFO* aos painéis da rede, que devem apresentar as mensagens desejadas afixadas. O *DMSConsole* após a seleção da fonte pelo usuário, realiza uma rotina a cada 2 minutos que envia uma mensagem *MSG INFO* á todos os painéis. Assim após inserir um dispositivo de armazenamento na porta USB do protótipo, é feito através da página de Seleção de Fontes a importação do ficheiro contendo o seguinte código XML:

```
1 <GID version="1.1">  
2   <Group ID="1">  
3     <Line ID="1" Name="Teste">
```

```
4 <Destination ID="1" Name="Teste" Return="1">
5 <Panel Lines="16" Columns="112">
6 <Page ID="1" SBC="0" SBNAndC="2" PT="15">
7 <Number Effect="None" Align="Left" Font="16x9">
8 1
9 </Number>
10 <Text Effect="None" Align="Left" Font="17x11">
11 ROTATE
12 </Text>
13 </Page>
14 </Panel>
15 </Destination>
16 </Line>
17 </Group>
18 </GID>
```

Após a importação do arquivo Xml(4.2), o painel de teste apresenta imediatamente a informação desejada, como pode ser visto na Figura (40).



**Figura 40: Painel Eletrônico da Rede de Indicadores de Destino - Sem Efeito de Rola-gem**

**Fonte: Aatoria Própria**

Para testar a resposta do painel a modificação na mensagem *MSG INFO* enviada, um novo arquivo fonte foi criado, semelhante ao utilizado anteriormente, porém com um efeito de rolagem no campo de texto do painel. O Resultado foi a exibição de um texto que se deslocava através do painel, como pode ser visto na Figura (41), o que comprova o funcionamento do console dentro dos casos de testes.



**Figura 41: Painel Eletrônico da Rede de Indicadores de Destino - Com Efeito Rolagem**  
Fonte: Aatoria Própria

A usabilidade da interface usuário-console foi testada com auxílio do próprio departamento de desenvolvimento da DMS, requisitando que 3 voluntários do departamento utilizassem a navegação e todos os recursos de configuração e de criação de fontes. Tendo em vista que estes 3 voluntários participaram do desenvolvimento do antigo modelo SICON II, e por isso já conheciam as necessidades que uma interface de console para indicadores de destino possui.

Todos os 3 voluntários elogiaram a disposição dos botões de interação e a legibilidade das informações exibidas na tela do *display*. Também houve uma boa avaliação em relação à facilidade e fluidez que a interface do console ofereceu. O único problema apresentado foi o delay dos botões *touch* apresentados. Principal-

---

mente durante o uso do teclado virtual, o que dificulta o uso do editor de fontes do DMConsole.

O problema de delay foi analisado, e avaliado como oriundo da sensibilidade reduzida do display Touchscreen. Já que mesmo adequadamente configurado, segundo o próprio manual do fabricante, quando há uma interação de toque do usuário o módulo, ocorre aleatoriamente um retardo na notificação ao dispositivo mestre sobre o evento. O que portanto acaba provocando este delay.



## 5 CONCLUSÃO

Ao longo deste trabalho foram apresentados os conceitos fundamentais acerca do funcionamento e funcionalidade do sistema de indicadores de destino, e consequente as vantagens de uma modernização por meio de um novo modelo de console para este sistema. Este novo console dotado de mais recursos de conectividade e interação com o usuário, desde *software* ao *hardware* foram aqui desenvolvidos. O *software* atendeu as necessidades do projeto e forneceu novas funcionalidades ao console, mantendo ainda o desempenho da aplicação. A interface de usuário, parte constituinte do *software*, se apresentou fluida, prática e amigável, tornando o console ainda mais útil e agradável ao usuário do que a antiga versão SICON II. Já o *hardware* apresentou-se bastante robusto e compacto, sendo todo o console constituído de apenas 3 PCBs; o Raspberry Pi, o módulo *Display Touch Screen* e a placa de integração. A placa de integração foi a única PCB desenvolvida neste projeto, sendo que ela reuniu na mesma placa o conversor UART RS485, o conversor UART RS232 e ainda um conversor CC-CC chaveado que alimenta todo o console. O projeto do novo console culminou na montagem do protótipo, o qual recebeu o nome de SICON III. Este protótipo montado em uma caixa de plástico adequada a instalação do console no painel do autocarro, se apresentou como um produto com bom acabamento e bem montado. Quando testado o protótipo funcionou perfeitamente, realizando a configuração do painel de teste e afixando adequadamente mensagens no mesmo. Ao fim foi desenvolvido um bom produto, dentro das necessidades e além das expectativas, e que possivelmente deve se tornar a próxima novidade na linha de indicadores de destino da DMS.



## REFERÊNCIAS

ADAFRUIT, Industries. **Introducing the Raspberry Pi Zero**. [S.l.], 2018.

BRAY, Tim; PAOLI, Jean; SPERBERG-MCQUEEN, C Michael; MALER, Eve; YERGEAU, François. Extensible markup language. **World Wide Web Journal**, v. 2, n. 4, p. 27–66, 1997.

CP. **CP, Comboios de Portugal**. 2018. <https://www.cp.pt/institucional/pt/empresa>. Accessed: 2018-05-15.

CUNHA, Liliana; LACOMBLEZ, Marianne. Mudanças no sector dos transportes em portugal: que caminhos para a actividade de serviço público e para a preservação do interesse geral?: estudos de caso. 2006.

DEBIAN, GNU. **Linux**. 2005.

DMS. Consola de indicador de destino - protocolo de comunicação. Rua de Cidres, nº1444, 4455-442 Perafita (Matosinhos), Portugal, 2004. SICON II - DOCIE05004-00.

DMS. Displays & mobility solutions. Rua de Cidres, nº1444, 4455-442 Perafita Matosinhos, Portugal, 2018. Accessed: 2018-05-15.

MURATA, Power Solutions. Oki-78sr series, non-isolated switching regulator dc-dc. **JP: Data Sheet**, 2010.

ORMAN, Levent V. Public information systems. **The Information Society**, Routledge, v. 6, n. 1-2, p. 69–76, 1989. Disponível em: <<https://doi.org/10.1080/01972243.1989.9960070>>.

PIGPIO. **The PiGPIO library**. [S.l.], 2018.

PRIYANKA, D.; DHARANI, K.; ANIRUDH, C.; AKSHAY, K.; SUNIL, M. P.; HARIPRASAD, S. A. Traffic light and sign detection for autonomous land vehicle using raspberry pi. In: **2017 International Conference on Inventive Computing and Informatics (ICICI)**. [S.l.: s.n.], 2017. p. 160–164.

SPARKFUN, Electronics. **GPIO - Raspberry Pi Zero v1.3**. 1.3. ed. [S.l.], 2018.

STCP. **Sociedade de Transportes Colectivos do Porto**. 2018. <http://www.stcp.pt/pt/comunicados-de-imprensa/>. Accessed: 2018-05-15.

TEXAS, Instruments. The rs-485 design guide. **USA: Texas Instruments**, 2008.

TEXAS, Instruments. Low-power rs-485 transceiver. **USA: Texas**, 2012.

TEXAS, Instruments. Max3232 3-v to 5.5-v multichannel rs-232. **USA: Texas**, 2012.

---

TSO, F. P.; WHITE, D. R.; JOUET, S.; SINGER, J.; PEZAROS, D. P. The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures. In: **2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops**. [S.l.: s.n.], 2013. p. 108–112. ISSN 1545-0678.