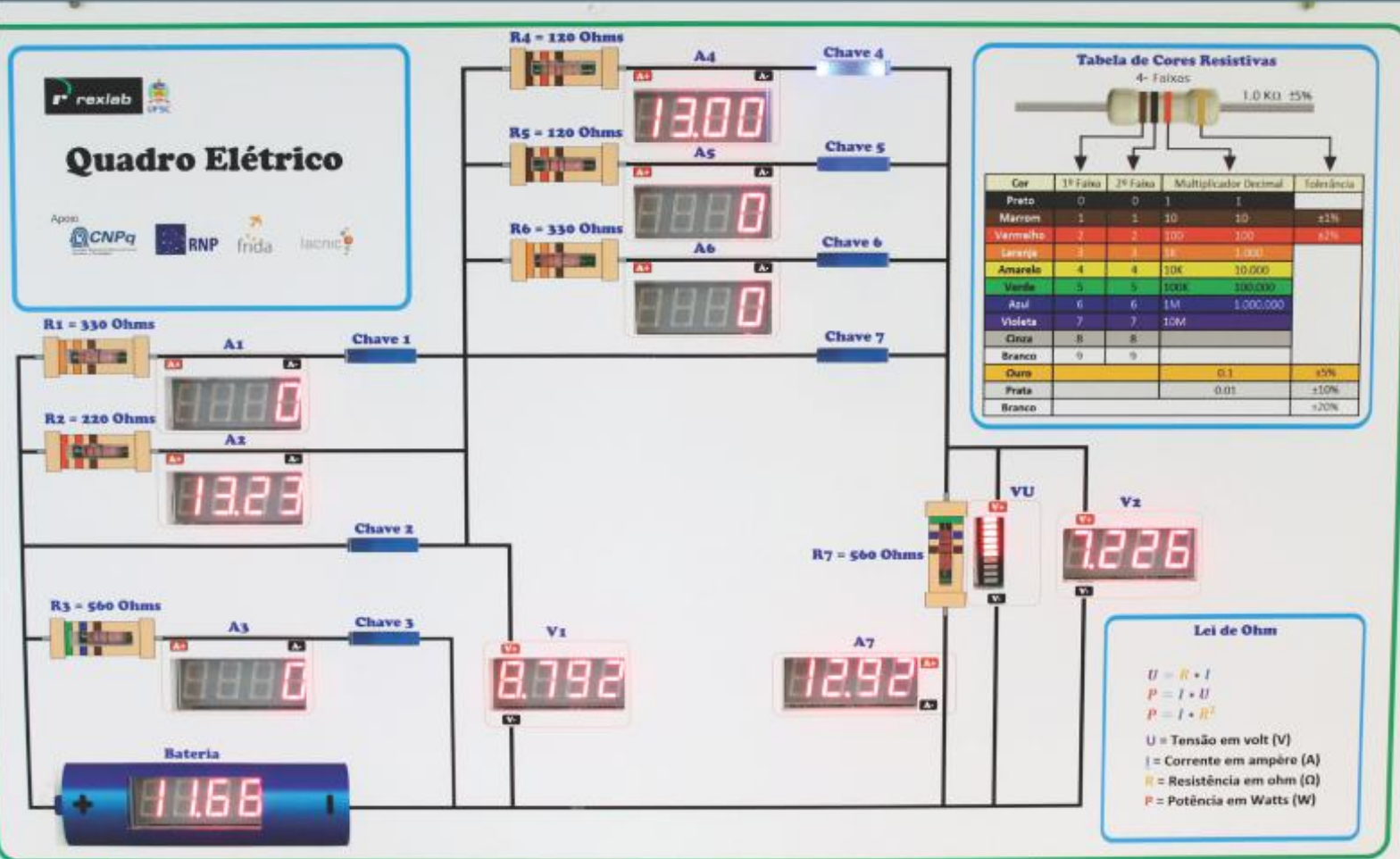


Manual Técnico

Painel Elétrico CC



Experimentação Remota Móvel para o Ensino Básico e Superior

Guia técnico do Experimento Remoto Painel Elétrico CC:
Experimentação Remota Móvel para a Educação Básica e Superior
Este guia, cada capítulo e suas imagens estão licenciados sob a licença
Creative Commons
Rua Pedro João Pereira, 150, Mato Alto – CEP 88900-000
<http://rexlab.ufsc.br/>
rexlabufsc@gmail.com

Edição

Juarez Bento da Silva

Prefácio

João Bosco da Mota Alves

Elaboração

Juarez Bento da Silva

João Paulo Cardoso de Lima

José Pedro Schardosim Simão

Lucas Mellos Carlos

Revisão

Simone Meister Sommer Bilessimo

Isabela Nardi da Silva

Editoria de arte, projeto gráfico e capa

Isabela Nardi da Silva

Ilustrações

Alex Moretti

Este guia, cada capítulo e suas imagens estão licenciados sob a licença Creative Commons - Atribuição-NãoComercial-Sem Derivados 4.0 Internacional. Uma cópia desta licença pode ser visualizada em [http://creativecommons.org/licenses/by-nc-nd/](http://creativecommons.org/licenses/by-nc-nd/4.0/)

Ela define que este manual é livre para reprodução e distribuição, porém sempre deve ser citado o autor. Não deve ser usado para fins comerciais ou financeiros e não é permitido qualquer trabalho derivado. Se você quiser fazer algum dos itens citados como não permitidos, favor entrar em contato com os organizadores do manual.

O download em edição eletrônica desta obra pode ser encontrado em <http://www.rexlab.ufsc.br>.



Guia Técnico do Experimento Remoto Painel Elétrico CC:
Experimentação Remota Móvel para a Educação Básica e Superior / obra coletiva concebida, desenvolvida e produzida pelo Laboratório de Experimentação Remota (RExLab)

Araranguá - SC, Brasil, 2016

Experimento Remoto Painel Elétrico CC

Apresentação

O experimento painel CC tem como objetivo auxiliar os estudantes do Ensino Médio e do Ensino Superior a efetuar práticas relacionadas à associação de resistores (série, paralela e mista), bem como, a análise de circuitos eletrônicos em corrente contínua usando as leis básicas de análise de circuitos: Lei de Ohm, Lei de Kirchhoff das Tensões ou Malhas e a Lei de Kirchhoff das Correntes ou Nós, a partir do uso de um enfoque prático da aplicação destas leis.

Este experimento foi desenvolvido no âmbito do O GT-MRE(Grupo de Trabalho em Experimentação Remota Móvel), com o objetivo de ser disponibilizado o seu acesso por meio de dispositivos móveis e dispositivos convencionais.

Arquitetura

O dispositivo está implementado a partir da estrutura padrão de hardware e software básico. Na Figura 1 pode ser visualizado o diagrama da arquitetura do experimento remoto.

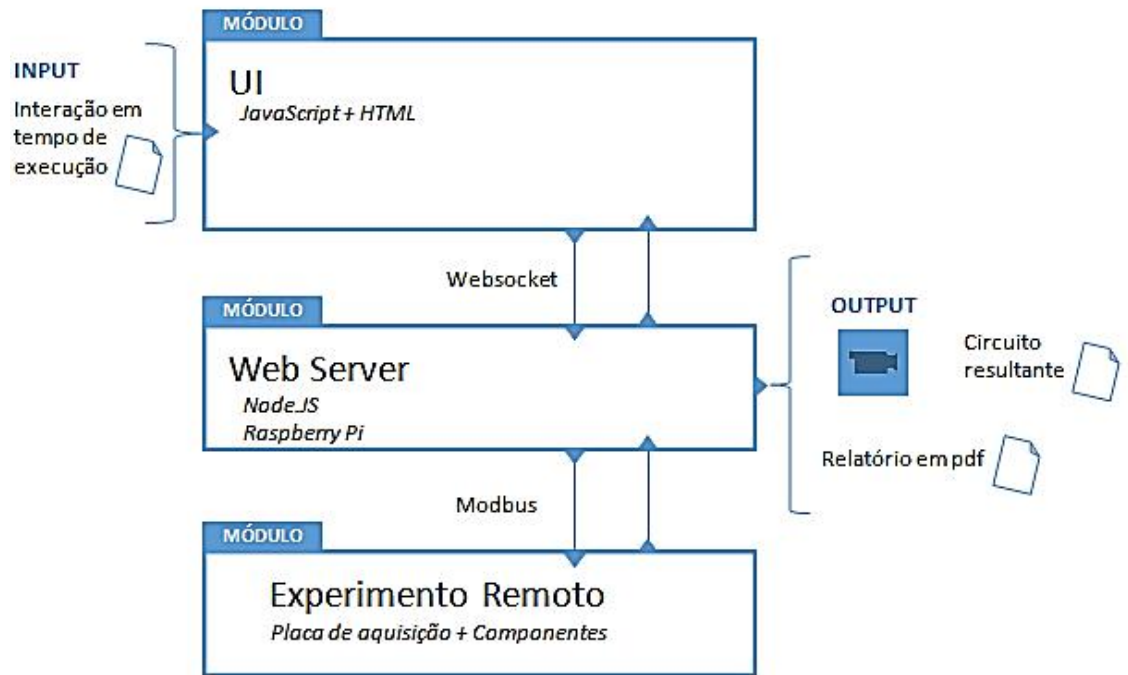


Figura 1 - Arquitetura do experimento remoto: Painel CC

Interface de Usuário (UI)

O experimento está disponível no sistema de gerenciamento RELLE (*Remote Labs Learning Environment*), que provê uma série de funcionalidades necessárias para o gerenciamento de experimentos remotos.

A interface de acesso ao experimento foi desenvolvida utilizando HTML juntamente com o *framework front-end Bootstrap*, o mesmo traz uma série de componentes prontos para o desenvolvimento além de prover tratamento para diferente tipos de resoluções de telas. Também é utilizado a biblioteca jQuery que traz uma série de funções *JavaScript* que simplificam o desenvolvimento.

A Figura 2 mostra como está disposto o experimento Painel Elétrico CC no Ambiente de Aprendizagem com Experimentos Remotos, ou RELLE, (do inglês *Remote Labs Learning Environment*, pode ser acessado em <http://relle.ufsc.br>).

Painel Elétrico CC

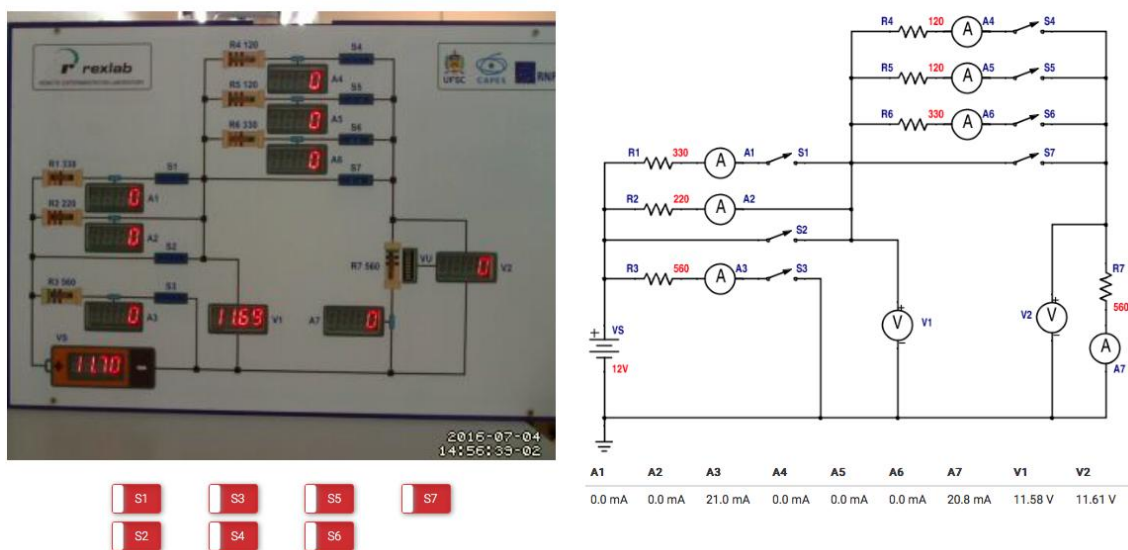


Figura 2 - Interface do usuário no RELLE

Web Server

Atualmente, há uma ampla gama de bibliotecas e frameworks para construção de serviços web. Apesar de serviços baseados em HTTP predominarem a Internet, o uso do protocolo *WebSocket* é uma tendência em aplicações corporativas de grande porte. Uma das plataformas para desenvolvimento web para construção de serviços baseados em *WebSocket* é o *framework NodeJS*.

O *NodeJS* permite construir aplicações de servidor e de rede facilmente escaláveis. Ele é composto por um ambiente de execução multiplataforma e de código fonte aberto que interpreta códigos de aplicações escritas em *Javascript*. O *NodeJS* usa um modelo orientado a evento, com operações de entrada e saída não bloqueantes. Por este motivo, ele é ideal para aplicações em tempo real com troca intensa de dados entre dispositivos distribuídos.

A API para acesso às funcionalidades do *SmartDevice* contém funções vinculadas à *listeners*, comuns ao paradigma de orientação a eventos. Este módulo usa a biblioteca *Socket.io* e é o ponto de partida da aplicação,

onde o servidor é iniciado e eventos são vinculados. O Socket.io é composto por dois componentes: servidor e cliente, ao qual usa principalmente o protocolo *WebSocket*, e *polling* HTTP como compatibilidade reversa.

A autorização de sessão no *SmartDevice* garante a integridade do acesso exclusivo, já que o dispositivo exposto como um serviço pode ser utilizado concorrentemente por outro cliente. Apesar de algumas funcionalidades poderem ser utilizadas no modo observador, como consultar o estado das chaves e metadados, as funcionalidades de controle necessitam de consulta ao sistema de fila.

O sistema de fila, ou mesmo agendamento, pode ser externo ou interno ao *SmartDevice*. O primeiro é baseado em um *token* de autenticação provido pelo usuário e validado pelo *SmartDevice*. As implementações dos experimentos de física exemplificam o uso do sistema de reserva externo (próprio do RELLE). Já o controle de acesso no próprio *SmartDevice* é exemplificado pela implementação do Laboratório de desenvolvimento em Arduino, pois neste encontra-se um modelo de acesso diferente dos anteriores.

O código fonte desenvolvido para comunicação serial e gerência dos sensores e atuadores são complementos para o *NodeJS* escritos em C++. Estes complementos são objetos compartilhados de vínculo dinâmico que pretendem dar suporte a códigos nativos, rapidez e portabilidade. Esses objetos compõem a abstração de cada experimento físico, que é representado por métodos e atributos intrínsecos a cada um. Por exemplo, são definidos os métodos de “*get*” e “*set*” para saídas digitais, “*get*” para valores de sensores, “*get*” e “*set*” para calibragem e configuração dos sensores.

O dispositivo central do experimento é o servidor de laboratório, que na plataforma desenvolvida pelo GT-MRE a escolha recaiu sobre o RaspberryPi¹ (Figura 3) modelo B+, que tem como principal função intermediar os acessos aos demais dispositivos de hardware dos experimentos via rede.

O servidor de laboratório (SL) tem função prover interfaceamento e gerenciamento para a conexão entre a rede (web) e a “placa de aquisição e controle” (PAC). O SL acessa a PAC para coletar os dados dos sensores ou

¹ O RaspberryPi é um computador baseado em um system on a chip (SoC) Broadcom BCM2835, que inclui um processador ARM1176JZFS rodando a 700 MHz, GPU VideoCore IV, e 512 MB de memória RAM em sua última revisão. O Raspberry Pi foi desenvolvido no Reino Unido pela Fundação RaspberryPi.

para enviar comandos para os atuadores, essa comunicação é feita via porta UART(Universal asynchronousReceiver/Transmitter) que se comunica via protocolo MODBUS2.



Figura 3 - Raspberry Pi, Model B+

API WebSocket

Os componentes da aplicação são suficientemente leves para serem executados por uma placa Raspberry Pi ou outro computador Linux de baixo custo. Um dos componentes, a API *WebSocket*, oferece uma interface aos sensores e atuadores na estrutura de um serviço web. A aplicação não requer alto uso da memória e pode ser utilizada em qualquer sistema Linux.

O resultado é uma arquitetura fracamente acoplada, adotada pelo GT-MRE, que habilita o compartilhamento dos experimentos em outras plataformas. Esse paradigma, chamado de SmartDevices³ já é utilizado no projeto Go-Lab⁴, no qual estão bem destacadas aplicações clientes e servidor, e fornecem interfaces bem definidas entre o usuário e o sistema.

²Modbus é um protocolo de comunicação de dados utilizado em sistemas de automação industrial. É um dos protocolos mais utilizados em redes de Controladores lógicos programáveis (PLC) para aquisição de sinais (0 ou 1) de instrumentos e comandar atuadores. É de utilização livre e sem taxas de licenciamento.

³ DOI: 10.1109/REV.2015.7087292

⁴<http://www.go-lab-project.eu/>

Os tópicos seguintes apresentam com mais detalhes aspectos do serviço web utilizado no servidor de experimento, bem como as funcionalidades internas e as motivações para o uso de certos protocolos, padrões e ferramentas de desenvolvimento, conforme a Figura 4.

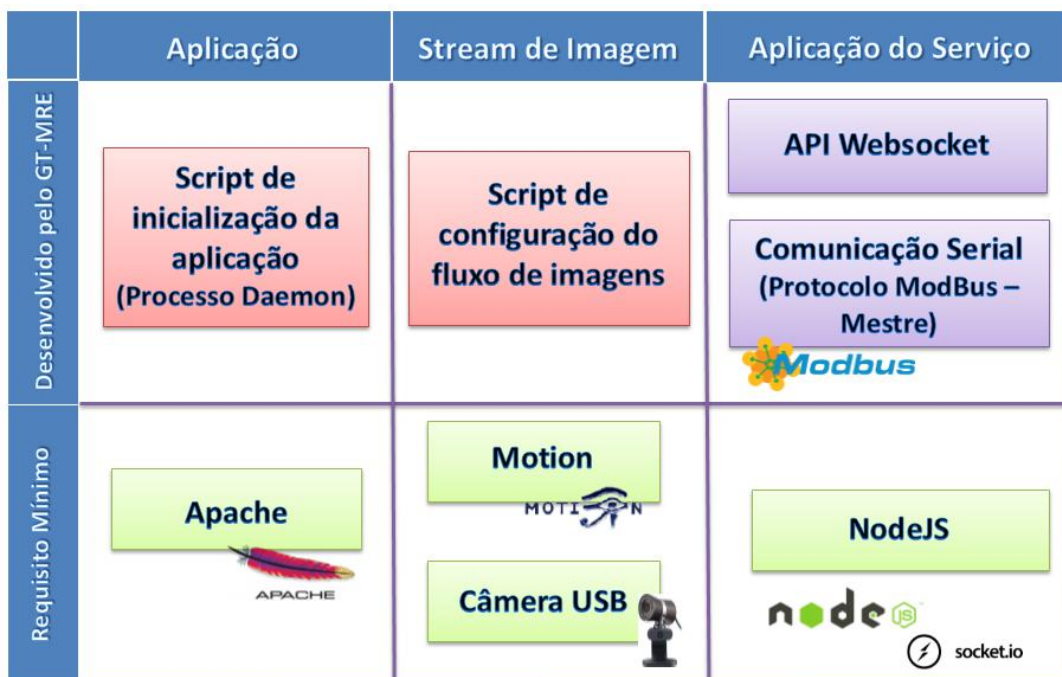


Figura 4 - Esquema de aplicação embarcada. Fonte: GT-MRE.

Controle e monitoramento do experimento

OSmartDevice é capaz de comunicar-se com sensores através do barramento serial (Porta UART). Ao invés de usar o protocolo serial em sua forma bruta, optamos por incluir o protocolo Modbus na camada de aplicação para identificação de erros, endereçamento e controle de colisão. Conectados ao mesmo barramento (rede), cada sistema embarcado, responsável por um ou mais sensores ou atuadores, é um dispositivo escravo que responde às requisições da aplicação que é executada no Raspberry Pi.

Um dos módulos desenvolvidos para aplicação é responsável pelo serviço de fila externo ou interno, sendo possível acoplar o serviço de fila provido pelo RELLE ou habilitar serviços internos. No primeiro caso, a aplicação usa a lógica necessária para validação de *token* de sessão enviado

pelo cliente. Na segunda, todo processo realizado pela web API de fila é realizado pelo SmartDevice.

Acesso à web API pelo cliente

A Figura 6 apresenta o esquema de comunicação no uso da API desenvolvido para o serviço/protótipo.

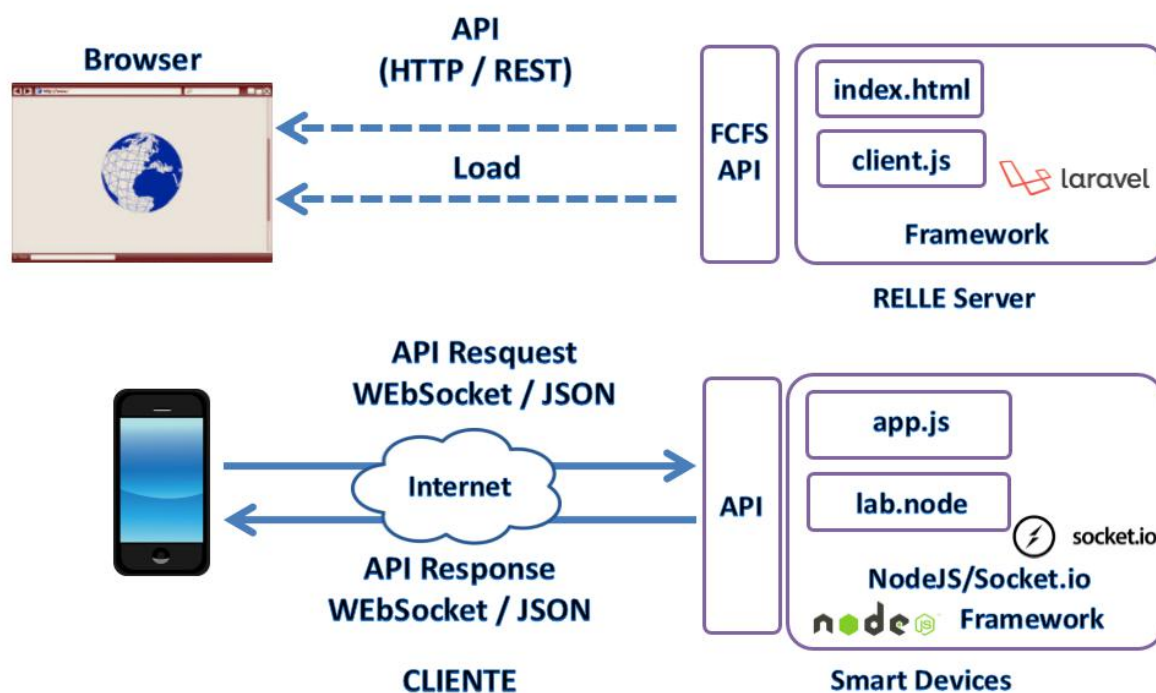


Figura 5 - Esquema de comunicação crossdomain no uso da API desenvolvida pelo GT-MRE.

O cliente web disponibilizado pelo sistema RELLE é composto por um arquivo html, css e javascript diferentes para cada experimento. O RELLE provê uma página comum para cada experimento onde carrega os dados que foram inseridos no momento da publicação do experimento (armazenados numa base de dados). Por exemplo, o experimento de ID 1 é acessível pela URL “relle.ufsc.br/labs/1” pelo método GET e contém suas informações dentro do layout padrão do sistema. A partir do botão “Acessar” é possível disparar um evento para comunicação com a Web API FCFS (first-come firstserved).

Ao obter a permissão no navegador, o cliente navegador poderá carregar os arquivos (html, css e js), pois a API já tem o seu token de sessão

como usuário sendo servido. Após carregar o cliente para o SmartDevice (client.js), uma conexão WebSocket com este dispositivo é estabelecida.

Streaming de imagens

No GT-MRE foi optado pelo uso de câmeras web com conexão USB devido ao baixo custo e a facilidade de aquisição. O mesmo computador embarcado utilizado para controle do experimento também é o responsável pelo gerenciamento e disponibilização do streaming no formato MJPEG (Motion JPEG). O MJPEG é um formato de compressão de vídeo na qual cada frame de vídeo é comprimido separadamente como uma imagem JPEG.

Visto que existem muitos servidores de streaming de código aberto, optou-se pelo Motion para explorar aspectos de leveza (utilização de poucos recursos) e configuração flexível. O Motion⁵ é um software escrito em C para sistemas Linux que usa a API de vídeo linux, e é capaz de detectar se uma parte significativa da imagem tem mudado. Algumas variáveis são ajustadas através de seu arquivo de configuração principal para adequar-se aos requisitos de nossa aplicação.

Atualmente, os principais navegadores do mercado como Firefox, Google Chrome e Safari já possuem o suporte nativo para o streaming MJPEG. Para clientes Android existem bibliotecas de código fonte aberto para incluir um visualizador MJPEG em aplicações de código nativo.

⁵<http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>

Experimento Remoto

O “Painel Elétrico CC” é composto de resistores, chaves, LEDs e duas fontes de alimentação. A partir das diversas combinações de acionamento das chaves inseridas no circuito elétrico é possível obter leituras de tensão e corrente em diversos pontos do circuito de acordo com as associações destes.

O dispositivo está implementado a partir da estrutura padrão de hardware e software básico. O circuito DC é composto por um painel acrílico onde estão instalados sete resistores que através das combinações de acionamentos das sete chaves distribuídas ao longo do circuito, permitem obter as diversas combinações para os circuitos estudados, de acordo com o objetivo do experimento. O experimento é mostrado na Figura 7.

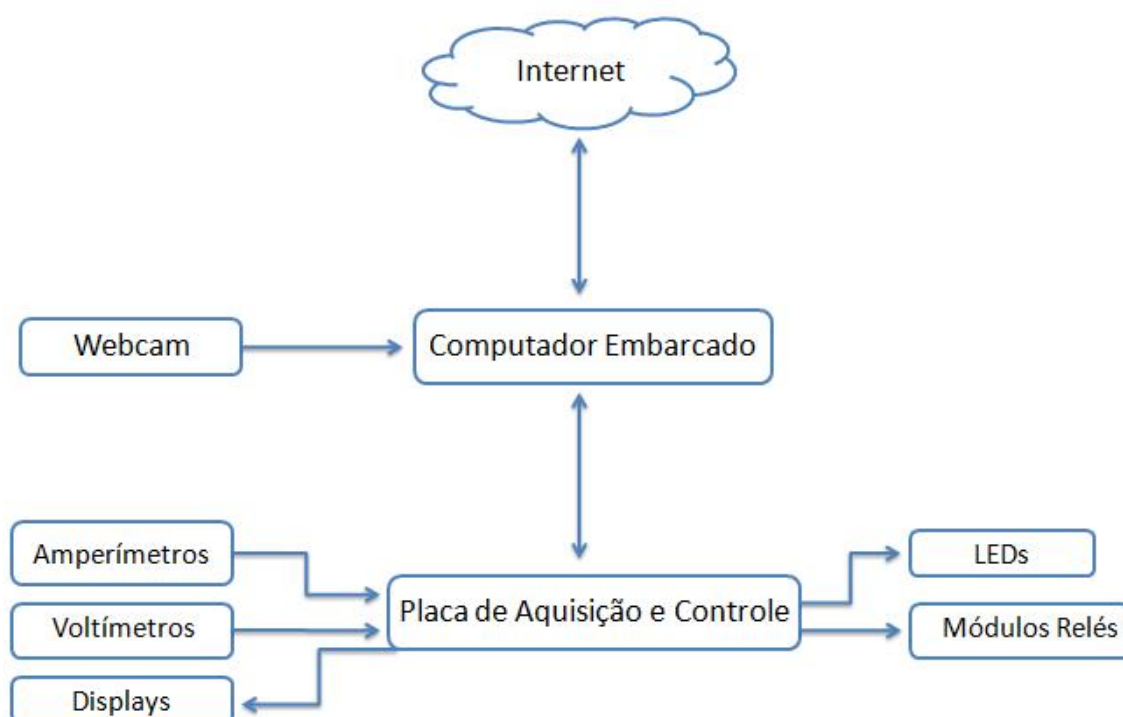


Figura 6 – Diagrama de blocos do experimento

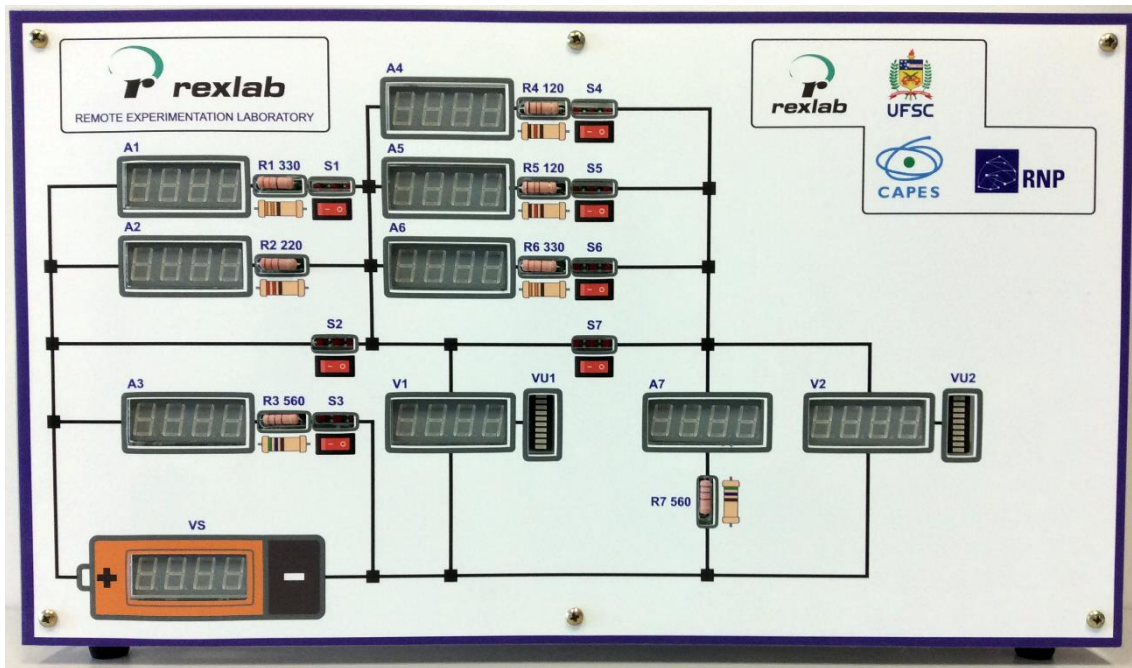


Figura 7 - Painel Elétrico CC

Abaixo, uma tabela com a lista de componentes utilizados no painel elétrico CC:

Quantidade	Componente
7	Chave
7	Resistor
7	LED
7	Amperímetro
3	Voltímetro
7	Módulo relé
10	Display 7 segmentos
1	Placa de aquisição e controle

Os módulos relés são dispositivos responsáveis por realizar a abertura e fechamento das chaves. A Figura 8 apresenta um modelo em 3D do seletor de tensão e do módulo do resistor, o esquema elétrico encontra-se disponível nos Apêndices.

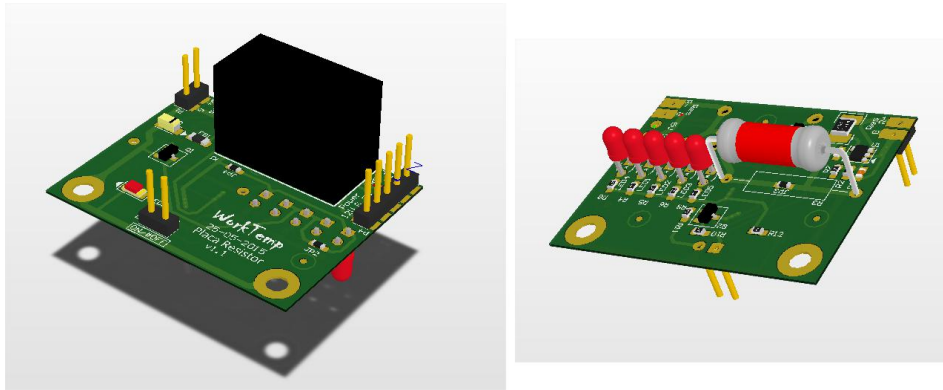


Figura 8 - Módulos relê e resistor em 3D

Os resistores utilizados no experimento são construídos em filme de carbono, potência de 3W e tolerância de 5%. O uso dessa potência é devido ao seu tamanho físico, a fim de proporcionar visualização mesmo através da câmera instalada para acompanhamento da experimentação.



Figura 9 - Resistor

Funcionamento dos multímetros e amperímetros

Estes dispositivos estão baseados no processador ARM Cortex-M3 LPC1752. A função de multímetro ou amperímetro é selecionada no dispositivo através de um jumper, que seleciona o que será exibido no display de 7 segmentos, pois, este trabalha tanto como voltímetro ou como amperímetro e nos dois casos os dados podem ser adquiridos via MODBUS.

Para leitura de corrente foi usado o sensor analógico de corrente, INA-169, indicado para medição de baixos valores de correntes DC, neste caso foi dimensionado para leitura de correntes máximas de 50mA.

O funcionamento deste circuito integrado consiste basicamente em converter a corrente para tensão a partir da leitura da “queda de tensão” em resistor “shunt”⁶, onde essa queda é referente a corrente que passa através

⁶O resistor shunt nada mais é que um resistor de baixo valor, ligado em série com a carga

dele. O INA-169 converte proporcionalmente a tensão de entrada em uma pequena corrente de saída. Na saída do INA-169 a corrente é convertida para tensão novamente através de um resistor externo (R_L) conectado ao GND do circuito. Isto é útil para podermos medir a corrente da linha através de um ADC ou outros dispositivos. Para leitura das tensões no circuito foi implementado um divisor resistivo com capacidade para leitura de tensões entre 0 a 15V em um microcontrolador (μC) de 3v3.

A medição deve ser no modo comum, ou seja, tanto o INA-169 como o aparelho em medição (carga ou load) devem ter um ground comum. A tensão de alimentação da carga e a tensão de alimentação do INA-169 podem ser entre 2v7 a 60v e são independentes entre si.

A Figura 10 apresenta o circuito de testes para o INA-169.

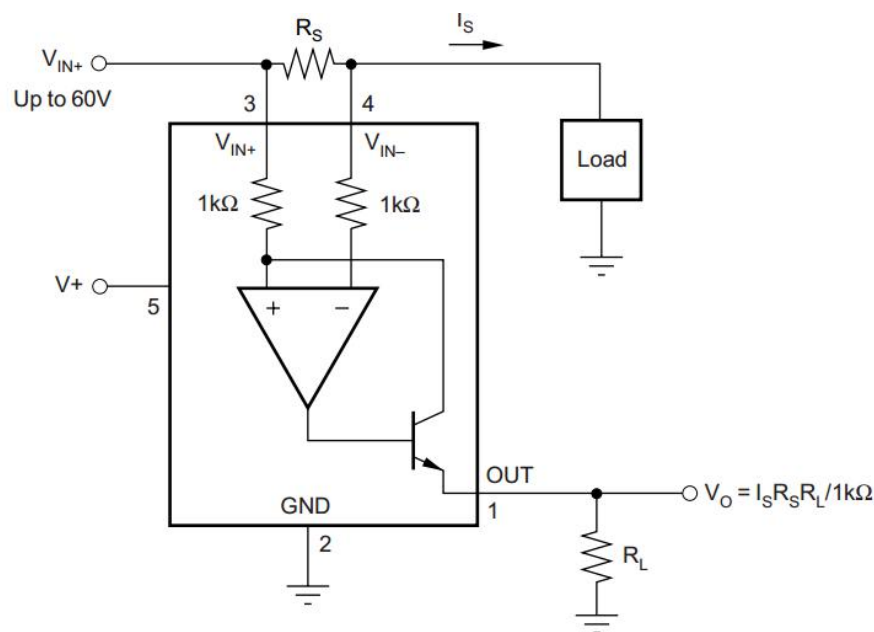


Figura 10 - Circuito de teste do INA-169.

A tensão diferencial no resistor de shunt deve ser no máximo de 0.5V, onde produz uma corrente de $500\mu A$, porém tensões acima disso não danificam o CI, embora saturam a tensão de saída. A medição diferencial deve ser unipolar, onde a tensão positiva deve ser conectada a entrada positiva do CI, caso ocorra a inversão, a corrente produzida será zero. Portanto, o INA-169

usada para fornecer um feedback (retorno) ao microcontrolador, com isso o microcontrolador identifica se a corrente fornecida está em conformidade (ou não) com o esperado

somente mede corrente DC, isto porque a entrada de tensão no Vin+ deve ser sempre maior que a entrada Vin-. O resistor shunt (R_s) deve ser colocado o mais próximo possível dos terminais do CI para que diminua o máximo a resistência em série com o R_s produzida pela trilha.

O valor do R_s depende da aplicação relacionando a precisão e a perda de tensão na linha de medição. R_s com valores mais altos são mais precisos em baixas correntes, porém em altas correntes há muita perda de tensão e muita potência é dissipada, neste caso devemos usar R_s com valores bem baixos. Geralmente é usado valor para R_s para fornecer uma escala completa onde a diferença de tensão varie na faixa entre 50mV a 100mV. Valores acima de 500mV o INA-169 satura e valores abaixo de 35mV o INA-169 começa a medir de forma errada. Usualmente utilizamos valores de R_s de 0.1, 1 e 10 ohms.

O valor do R_i é escolhido para uma escala completa de tensão de saída, como a impedância de saída do INA-169 é bem alta podemos usar um resistor com valor de até 100K com boa precisão de medição. Entretanto, o complemento do circuito, como um ADC, tem que ter a sua impedância de entrada maior que R_i , caso não for, é preciso usar um amplificador operacional configurado como buffer.

O valor de R_i converte a corrente interna do ina169 para tensão e também é responsável direto pelo ganho do circuito entre 1 a 100 vezes, observe as fórmulas a seguir.

- F1: $V_{out} = (I_s * R_s * R_i) / 1K$

- F2: $I_s = V_{out} / (R_s * R_i)$

Onde:

- V_{out} em V;
- I_s em A;
- R_s e R_i em Ω ;
- 1K constante do INA-169.

A tensão de saída (V_{out}) está relacionada diretamente com a tensão de alimentação do INA-169, ou seja, a tensão máxima que V_{out} pode alcançar é:

- F3: $V_{out_max} = V+ - 0.7 - (V_{in+} - V_{in-})$

Caso seja utilizado um ADC de 3v3 a tensão do INA-169 deverá ser maior que 3v3 para cobrir toda faixa de trabalho do ADC.

Utilizando os valores de R_s usuais a queda de tensão no R_s máximo é de 0.5V, tipicamente é de 0.1V, logo a relação de I_s com R_s baseado na lei de Ohm ($U=R*I$) é de:

$0.5/I_s=R_s$ (máximo)

Rs	I_{max}
0.	5A
1	
1	500mA
10	50mA

$0.1/I_s=R_s$ (Típico)

Rs	I_{max}
0.1	1A
1	100mA
10	10mA

Em relação ao experimento implementado, buscou-se medir uma corrente de 0 a 50mA em um ADC de 3v3 de 12bits de resolução. Uma vez que, o ADC é de 3V3 o ina169 deverá ser alimentado por 5V no mínimo (Fórmula 3) e a queda de tensão no resistor de shunt não deverá ser superior a 500mV, tipicamente será de 100mV. Uma vez que, a escala é de 0 a 50mA para valores de tensão de 0 a 3v3 estabeleceu-se o ponto central de 1.65 volts ($3v3/2$) que corresponde à para 25mA.

Efetuando a relação de R_i/R_s , aplicando a fórmula 1 obtêm-se:

- $V_{out} = (I_s * R_s * R_i) / 1$

- $1.65 = (0.025 * R_s * R_i) / 1$

Logo a relação entre os resistores é:

- $R_i = 66/R_s$

Considerando que os valores usuais do R_s são 0.1, 1 e 10 ohms obtém-se:

R_s	R_i
0.1	666K
1	66K
10	6K6

Para essa aplicação o valor de 0,1 ohm para R_s não é válido, pois, o valor máximo permitido para R_i é de 100K. Utilizando um resistor de carga R_i com valor comercial de 68K e o resistor de shunt R_s de 10hms e aplicando estes valores Fórmula 2 é possível calcular a valor da corrente.

- $I_s = V_{out}/(R_s * R_i)$
- $I_s = V_{out}/(68)$

Onde:

- V_{out} em V;
- I_s em Ω ;

A tensão V_{out} , corresponde à tensão a ser analisada, ou seja, a tensão que será aplicada na entrada do ADC. A sua precisão e escala depende do bit de conversão, ou bit de resolução do ADC e de sua tensão máxima de conversão. Uma vez que, uma tensão de 0v sempre resultará em um valor 0 no ADC, e a conversão de analógico para digital é linear, podemos estabelecer uma regra de três:

Tensão		Decimal
0	----->	0
U2	----->	d
vMax	----->	dMax

Onde:

- v_{Max} : Tensão máxima de conversão do ADC;
- d: Valor decimal lido pelo ADC;
- d_{Max} : $2 \exp(\text{bit conversão do ADC}) - 1$.

Embora a arquitetura ARM disponha de uma ampla gama de periféricos, nesta implementação foram utilizados somente os seguintes recursos:

- ADC0 canal 4: para leitura da tensão para o voltímetro;
- ADC0 canal 5: para leitura da tensão originada do conversor corrente de tensão para o amperímetro;
- GPIO: para leitura do MODBUS a ser acessado pelo mestre, acionamento do display e indicação do que será exibido no display;
- UART0: para depuração e comandos avançados;
- UART1: para comunicação MODBUS, via RS485.

Todos os multímetros e amperímetros estão conectados em um barramento RS485 que funciona em modo “escravo” no protocolo MODBUS. Cada multímetro tem um ID próprio para ser acessado pelo mestre, esse ID pode ser ajustado via jumpers.

O “recurso de hardware” contempla dois barramentos, ambos utilizando protocolo MODBUS. O primeiro barramento, é o barramento RS485, que se comunica com os multímetros em modo “mestre”, e fica periodicamente lendo os valores das tensões e das correntes em todos os dispositivos. O segundo barramento, é o barramento RS232, o MODBUS opera no modo escravo e aguarda a comunicação com o MSIP para fornecer os valores de leitura dos multímetros e amperímetros e também de outras funções. O MSIP dispõe de uma API que utiliza a UART do RaspberryPi, utilizando o protocolo MODBUS no modo mestre para acessar o recurso de hardware em busca dos dados à serem disponibilizados na WEB.

As Figuras 11 e 12 mostram os multímetros e amperímetros construídos e conectados, o diagrama do circuito pode ser conferido ainda nos Apêndices deste documento.

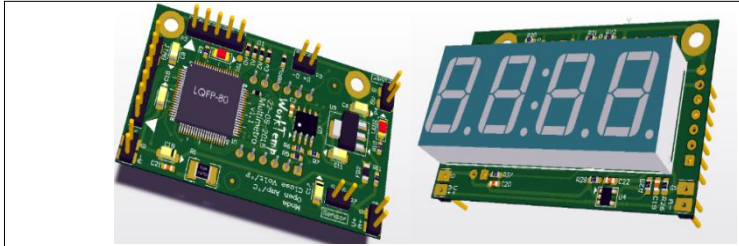


Figura 11 - Multímetros e amperímetros construídos.

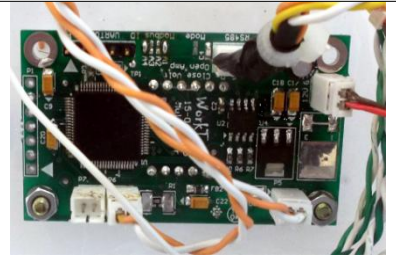


Figura 12- Multímetro conectado

Para a ligação entre as placas desenvolveu-se uma placa de união o que permite eliminar maioria dos fios e dando uma maior flexibilidade na criação de novos circuitos para projetos futuros usando os mesmos hardwares. A Figura 13 exhibe um diagrama em 3D da placa.

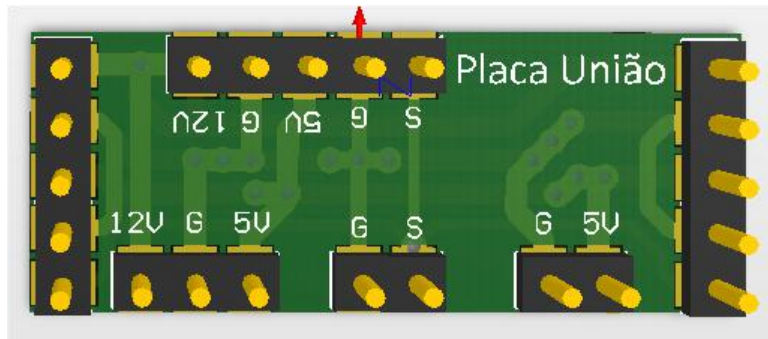


Figura 13 - Placa de ligação em 3D

Apêndices

Mapa de registradores

Todos os registradores são de 16 bits e os endereçamentos mostrados neste documento são em hexadecimais.

Registradores de Identificação		
Endereço	Tipo de Acesso	Descrição
0x0	Leitura	Identificador do modelo do aparelho. Valor em ASCII
0x1	Leitura	Versão do modelo do aparelho. Valor em ASCII
0x2	Leitura	Versão do firmware. Valor em ASCII. O formato da versão é x.y, porém o será enviado sem o ponto decimal. Exemplo: versão 1.0 será transmitido 10.

Registradores de Trabalho		
Endereço	Tipo de Acesso	Descrição
0x10	Leitura/escrita	Quantidade de multímetros para serem lidos no barramento rs485. Padrão é 9 e no máximo é 16. Os multímetros devem serem endereçados no barramento modbus de 1 a 16

Registradores de Saídas Digitais		
Endereço	Tipo de Acesso	Descrição
0x200	Leitura/escrita	O valor desse registrador espelha na saída digital do Recurso de Hardware. O bit0 para a o bit 0 da porta de saída e assim sucessivamente até o bit16

Registradores dos Reles		
Endereço	Tipo de Acesso	Descrição
0x300	Leitura/escrita	O valor desse registrador espelha o acionamento

		ou não dos reles do Recurso de Hardware. O bit0 para a o rele1 e assim sucessivamente até o bit16
--	--	---

Os multímetros no barramento RS485 devem ser endereçados (ID) do valor 1 em diante e nunca repetir o mesmo endereço para os multímetros. Os valores de amperagem e tensão são representados respectivamente em microampères e milivolts.

Registradores dos Multímetros		
Endereço	Tipo de Acesso	Descrição
0x400	Leitura	<p>Status de Comunicação</p> <ul style="list-style-type: none"> • 0: Sem comunicação com o dispositivo. O mesmo não está conectado, ou está desligado, ou não há dispositivo neste endereço. • 1: O multímetro recebeu uma função que não foi implementada; • 2: Foi acessado a um endereço de registrador inexistente; • 3: Foi tentado gravar um valor inválido no registrador do multímetro; • 4: Um irrecuperável erro ocorreu enquanto o multímetro estava tentando executar a ação solicitada; • 5: Comunicação estabelecida com sucesso.
0x401	Leitura	<p>Status do Multímetro com ID=1</p> <p>Função assumida do multímetro:</p> <ul style="list-style-type: none"> • Bit[4]: <ul style="list-style-type: none"> ○ 0: Voltímetro; ○ 1: Amperímetro. <p>Status do sensor:</p> <ul style="list-style-type: none"> • Bits[3:0]: <ul style="list-style-type: none"> ○ 0: Sinaliza que o multímetro está lendo pela

		<p>primeira vez o sensor. Isto somente acontece no momento que o multímetro é ligado.</p> <ul style="list-style-type: none"> ○ 1: O multímetro já contém o valor convertido; ○ 2: Sinaliza um erro, indica que o valor está abaixo da escala permitida pelo multímetro; ○ 3: Sinaliza um erro, indica que o valor está acima da escala permitida pelo multímetro;
0x402	Leitura	Word menos significativo do valor da amperagem ou voltímetro dependo da função assumida do Multímetro com ID=1
0x403	Leitura	Word mais significativo do valor da amperagem ou voltímetro dependo da função assumida do Multímetro com ID=1

0x404 = 0x400 porém para o multímetro com ID=2

0x405 = 0x401 porém para o multímetro com ID=2

0x406 = 0x402 porém para o multímetro com ID=2

0x407 = 0x403 porém para o multímetro com ID=2

E assim sucessivamente até o multímetro 16

Tutorial de reinicialização do experimento

Para reiniciar o experimento usa-se um terminal para conexão ssh, por exemplo o software PuTTY, o qual pode ser baixado pelo seguinte endereço:<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. Utilizando o PuTTY, basta inserir o endereço IP do experimento que se deseja reinicializar.

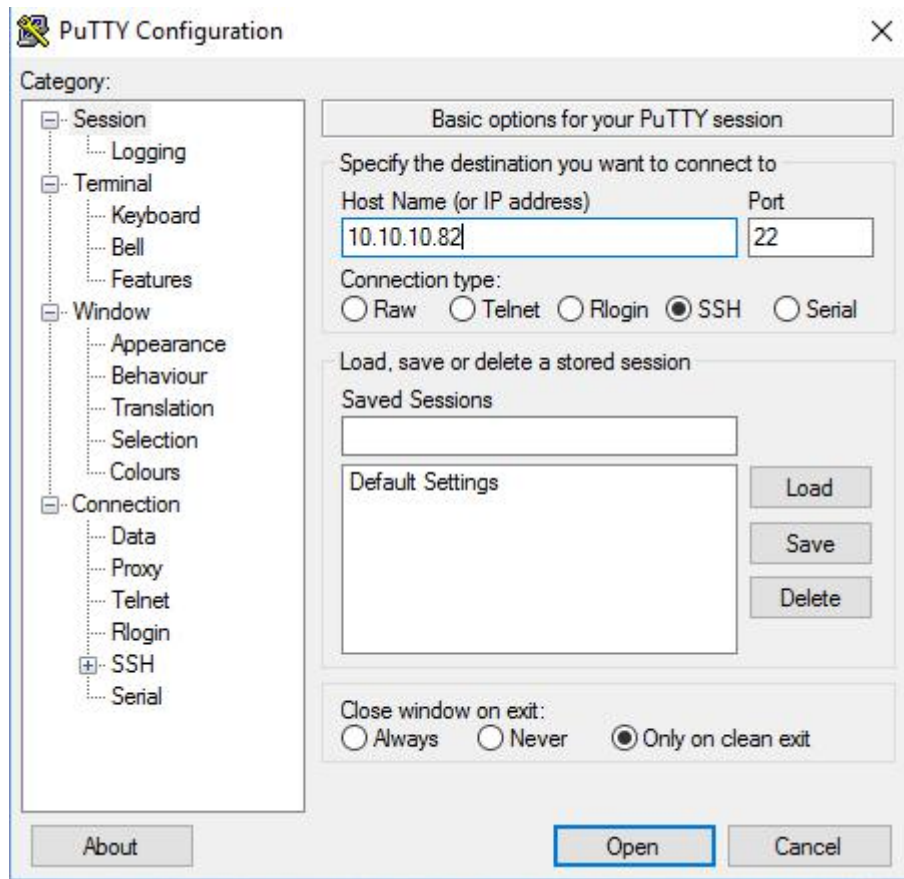


Figura 14 – PuTTY

Ao ativar a conexão será aberto um terminal (Figura 15), o qual solicitará um usuário (user) para autenticação. Recomenda-se autenticar com o usuário root, logo em seguida será requisitada a senha do computador embarcado. E por fim, para reiniciar o computador embarcado, digite o comando *reboot* no terminal.



Figura 15 - Terminal SSH com experimento

Verificação e reinício do serviço

Para verificar se os serviços do laboratório remoto estão rodando, basta usar o comando `ps -aux | grep node` que irá verificar os processos rodando referente ao servidor web Node.JS responsável por executar o serviço da aplicação. Caso o serviço esteja rodando, o resultado será algo similar a Figura 16 que exibe o usuário e número do processo em execução. Neste caso o processo PID 2434.

```
[root@raspberrypi:~# ps -aux | grep node
warning: bad ps syntax, perhaps a bogus '-'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
root    2434  0.1  9.7 1182324 43412 ?        Sl    May12 10:23 /usr/local/bin/node /home/conducao_app/apps.js
root    21479 0.0  0.3  3520  1740 pts/0    S+   14:17  0:00 grep node
```

Figura 16 - Verificação do serviço

Ações de iniciar, pausar ou verificar status do serviço podem serem executadas usando os comandos `servicedcapstart|stop|status`.

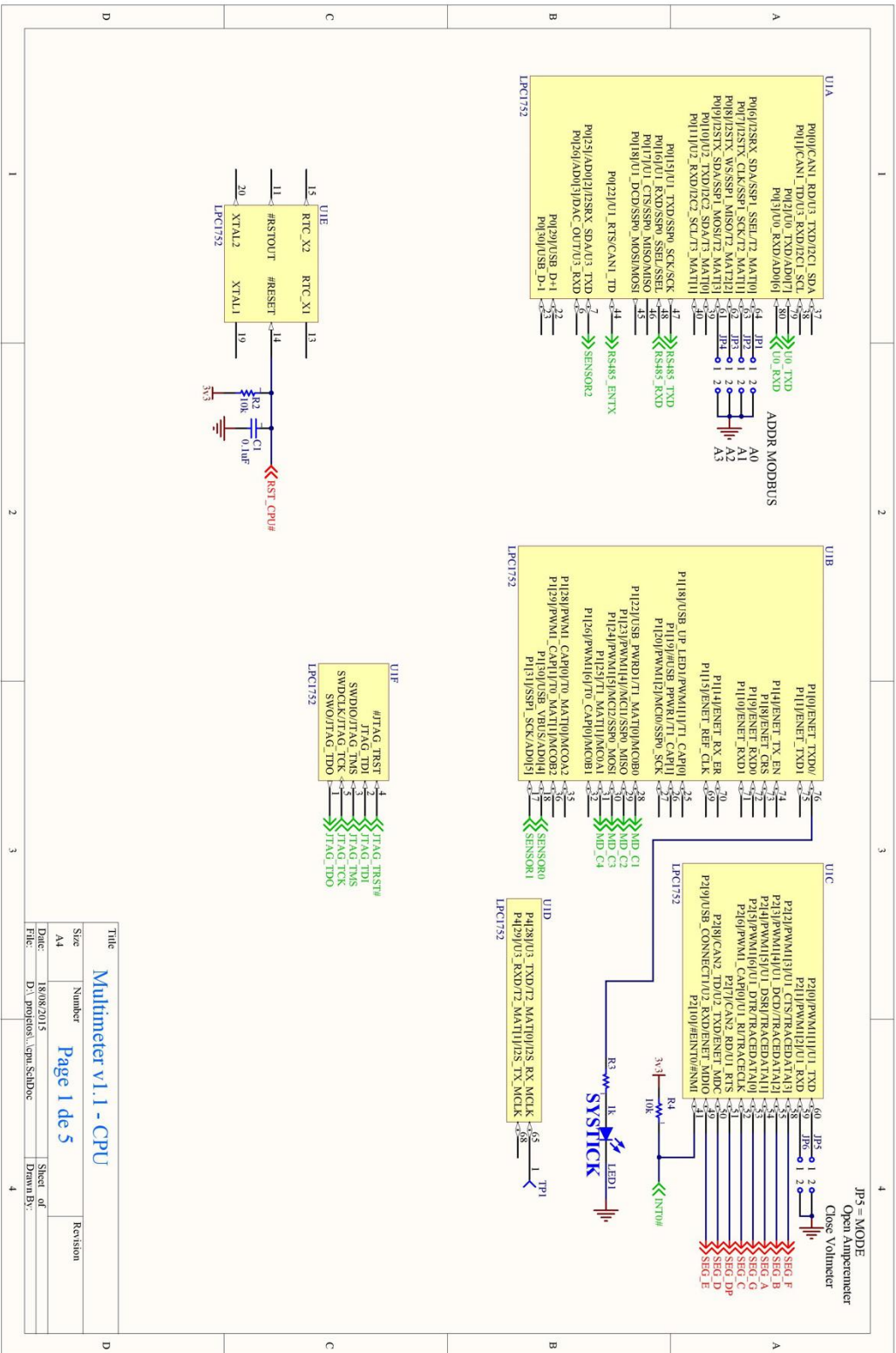
Manutenção do streaming de vídeo

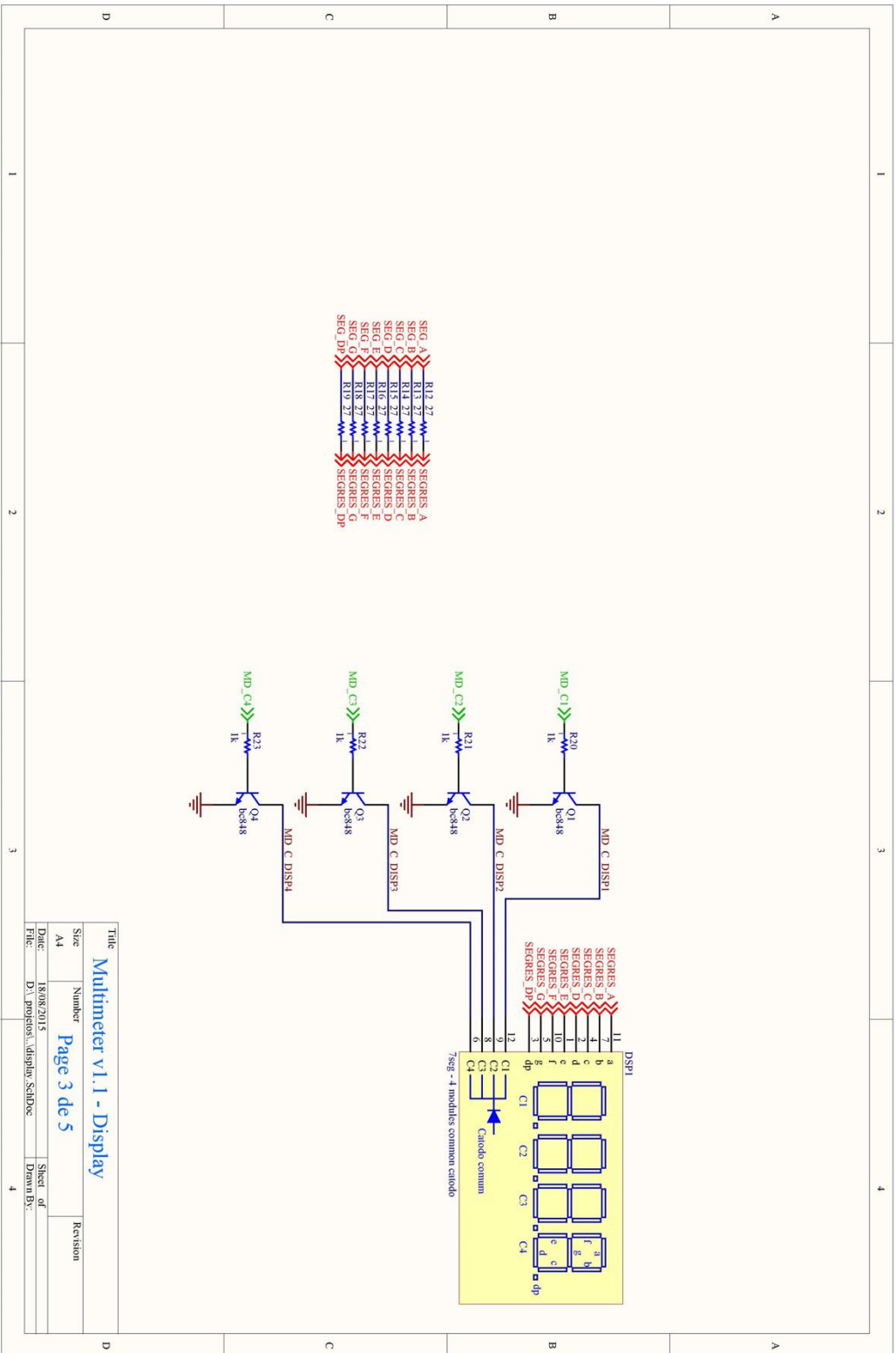
O vídeo é transmitido pelo software Motion. Para instalação do software pode-se fazer seu download via repositório através do comando `apt-get install motion` e acessar os arquivos de configurações `motion motion.conf`

através de algum editor de código no diretório */etc/default/motion* definindo o parâmetro *start_motion_daemon* para o valor *yes*.

As configurações relacionadas a qualidade da imagem e a transmissão ficam disponíveis no arquivo *motion.conf* no diretório */etc/motion/*. Ainda para início da transmissão dos parâmetros *deamone webcam_localhost* devem ser mudados para *one off*, respectivamente.

Diagramas dos multímetros





Title		Multimeter v1.1 - Display	
Size	Number	Sheet of	Revision
A4	18/08/2015	4	
Date:	D:\profesores\display_SchDoc	Drawn by:	
File:		4	

Diagrama da Placa Fonte

