UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ENGENHARIA INDUSTRIAL ELÉTRICA – ELETRÔNICA/TELECOMUNICAÇÕES

FÁBIO AUGUSTO ROSSETTI SOARES
RENATA LUMI SHIN-IKE

SISTEMA ELETRÔNICO DE ESTACIONAMENTO ROTATIVO

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2014

FÁBIO AUGUSTO ROSSETTI SOARES
RENATA LUMI SHIN-IKE

SISTEMA ELETRÔNICO DE ESTACIONAMENTO ROTATIVO

TRABALHO DE CONCLUSÃO DE CURSO

Trabalho de Conclusão de Curso de graduação do Curso de Engenharia Elétrica – Eletrônica/Telecomunicações da Universidade Tecnológica Federal do Paraná como requisito parcial para o título de Engenheiro Eletricista.

Orientador: Prof. M.Sc. Rafael Eleodoro de Góes

CURITIBA
2014

FÁBIO AUGUSTO ROSSETTI SOARES
RENATA LUMI SHIN-IKE

SISTEMA ELETRÔNICO DE ESTACIONAMENTO ROTATIVO

Este trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para a obtenção do título de Engenheiro em Engenharia Industrial Elétrica: Ênfase em Eletrônica/Telecomunicações pela Universidade Tecnológica Federal do Paraná.

Curitiba, 19 de março de 2014.

_____

Prof. Dr. Hilton José Silva Azevedo
Coordenador do Curso
Departamento Acadêmico de Eletrônica

_____

Prof. Dr. Dario Eduardo Amaral Dergint
Coordenador de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

_____

Prof. Dr. Rubens Alexandre de Faria

_____

Prof. M.Sc. Rafael Eleodoro de Góes

_____

Prof. Tasso Graeff Arnold

AGRADECIMENTOS

Agradeço aos meus pais, Leonilda e Marcos, por todo o tipo de suporte que me deram desde o início de minha vida.

Agradeço ao meu melhor amigo e irmão Marquinhos, pela amizade, conselhos e companheirismo.

Agradeço a minha namorada, Isabela, pelo apoio emocional, incentivo e me proporcionar diversos momentos de alegria durante esses últimos anos de engenharia.

Agradeço a todos meus amigos e colegas de faculdade, especialmente o Marco Antônio, pelo suporte técnico durante todo o curso.

Fábio Augusto Rossetti Soares

Agradeço ao meu pai Armando por todo apoio (muitas vezes mais em gestos do que em palavras) em minhas decisões, não só em relação aos estudos, mas também em toda minha vida.

Agradeço a minha mãe Nelci por ser meu maior exemplo de que fazer as coisas com amor nos torna mais feliz.

Agradeço ao meu irmão Vitor que desde pequena foi meu exemplo de pessoa inteligente, carismática e profissional.

Agradeço ao meu namorado Luigi por sempre estar ao meu lado, compreendendo e apoiando todas as minhas atividades, me fazendo rir e me sentir muito amada.

Agradeço aos meus amigos que me proporcionaram momentos inesquecíveis e que trouxeram muitos sentimentos e valores bons no meu crescimento como pessoa.

Agradeço a todos os professores que compartilharam conhecimentos, experiências e valores, sendo fundamentais para a nossa formação profissional.

Renata Lumi Shin-Ike

RESUMO

ROSSETTI SOARES, Fábio Augusto; SHIN-IKE, Renata Lumi. Sistema eletrônico de estacionamento rotativo 2014. 70 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Industrial
Elétrica com Ênfase em Eletrônica e Telecomunicações) – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

Na cidade de Curitiba, em regiões de alto tráfego de veículos, foi introduzido um sistema tarifário com o intuito de democratizar a utilização de vagas públicas. Estes locais foram demarcados como áreas de EstaR (Estacionamento Regulamentado). Contudo, o sistema utilizado não é prático ou cômodo para o usuário, uma vez que consiste em um cartões de papel onde o motorista assinala o horário em que estacionou, garantindo a sua permanência regularizada na vaga durante a hora seguinte. Este processo implica em alguns problemas, como o desperdício de um cartão ao assinalar a hora errada, pagar por uma hora completa mesmo permanecendo estacionada na vaga por alguns minutos, dificuldade para encontrar local de compra, ou agente de trânsito no momento necessário, além da comercialização ilegal dos blocos de EstaR. Esse trabalho descreve o estudo e o desenvolvimento de um sistema alternativo à utilização dos blocos de papel. Este sistema consiste em um dispositivo eletrônico contendo créditos em que, através de um botão, o usuário ativa o desconto de créditos ao estacionar em uma vaga de EstaR. Estes créditos são comprados através de um *website* e cada crédito equivale a um minuto de estacionamento regulamentado. A transferência para o dispositivo é feita pela porta USB de um computador. O sistema ainda possui algoritmos de criptografia para proteger as informações de crédito e evitar fraudes. A alternativa desenvolvida visa um sistema mais justo em que o usuário pague pelo tempo real estacionado, ao invés de pagar o valor de uma hora mesmo quando permanece por menos tempo na vaga.

Palavras-chave: EstaR. Estacionamento Regulamentado. Criptografia. Dispositivo embarcado.

ABSTRACT

ROSSETTI SOARES, Fábio Augusto; SHIN-IKE, Renata Lumi. Sistema eletrônico de estacionamento rotativo 2014. 70 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Industrial Elétrica com Ênfase em Eletrônica e Telecomunicações) – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

A tariff system was deployed in areas of high traffic in Curitiba in order to democratize the occupation of public parking places. These places are called EstaR (from portuguese "Estacionamento Regulamentado") areas. However, this system is not so practical or convenient for the users, since it consists on paper pads on which the driver marks the parked time, ensuring his stay is regularized over the next hour. This process involves some problems such as the wasting of card when marking the wrong time, the payment of an complete hour even if the driver parked for a few minutes, the difficulty of finding a seller of those paper pads and the illegal sale of EstaR cards. This paper describes the study and development of an alternative to the use of paper pads. This alternative consists on a system on which a electronic device contain credits. The user presses a button to activate the discount of credits whenever is parked on a EstaR area. These credits are purchased through a *website* and each credit is equivalent to a minute parked. They are passed to the device by an USB cable attached to a computer. This system also has encryption algorithms to protect the credit information. This alternative seeks a fairer system where the user pays the actual parked time instead of paying the price for an hour even when he doesn't stays for this long.

Palavras-chave: EstaR. Regulated Parking. Encryption. Electronic device.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

EstaR - Estacionamento Regulamentado

SETRAN - Secretaria Municipal de Trânsito

RISC - Computador com um Conjunto Reduzido de Instruções (do original *Reduced Instruction Set Computer*)

IDE - Ambiente Integrado de Desenvolvimento (do original *Integrated Development Environment*)

USB –Barramento serial universal (do original *Universal Serial Bus*)

OLED - Diodo emissor de luz orgânico (do original *Organic Light-emitting Diode*)

RTC - Relógio de Tempo Real (Real-time clock)

UART - Universal Asynchronous Receiver/Transmitter

DPDT – Polo duplo, curso duplo (do original *Double Pole, Double Throw*)

SCL - *Serial Clock.*

SDA – *Serial Data.*

ACK – *Acknowledge.*

CI – Circuito Integrado.

AES – Padrão de criptografia avançada (do original *Advanced Encryption Standard)*

DES – Padrão de criptografia de informação (do original *Data Encryption Standard).*

API – Interface de programção de aplicativos (do original *Application Programming Interface*).

EEPROM – *Electrically Erasable Programmable Read Only Memory.*

RoHS – Restrição de Certas Substâncias Perigosas (do original *Resctriction of Certain Hazardous Substances*)

SUMÁRIO

1 INTRODUÇÃO

Com a crescente frota de automóveis nas áreas urbanas tornou-se necessária a realização de um sistema de democratização do espaço público utilizado como estacionamento. Em Curitiba, a solução adotada foi a utilização de blocos de cartões pagos (EstaR), em que o usuário registra a hora de chegada e posiciona o papel no painel do veículo, sinalizando sua estadia regularizada naquela vaga durante a próxima hora. De acordo com o prazo estabelecido nas placas de sinalização, o motorista pode utilizar mais de um cartão para a mesma vaga, possibilitando um período maior de estacionamento regulamentado.

Este trabalho tem como objetivo o desenvolvimento de uma solução tecnológica para o tratamento do estacionamento regulamentado. Ele consiste na construção de um dispositivo embarcado portátil de baixo consumo, contendo um visor e botões para interface com os usuários (motorista e fiscais) e que regulariza o estacionamento na vaga, descontando créditos, minuto a minuto, até o limite de tempo permitido no local.

1.1 JUSTIFICATIVA

A utilização do sistema de blocos de cartões consiste em o motorista destacar um cartão ao estacionar o carro, preencher o mês, dia, hora e minuto em que estacionou e deixar o cartão no painel do veículo de modo que fique visível para os Agentes de Trânsito da SETRAN que estejam atuando na operação do EstaR. Com isso, o estacionamento está regulamentado por uma hora a partir do horário assinalado. Os locais onde é necessária a utilização do cartão são sinalizados por placas de sinalização contendo o tempo máximo de permânencia na vaga, proporcionando, assim, a rotatividade. O prazo de permanência para cada vaga é definido pela placa de sinalização, podendo ser de 1 hora, 2 horas e 3 horas, sendo um cartão para cada hora.

Foi realizada uma pesquisa com 348 pessoas sobre o EstaR, e concluiu-se que este sistema não é prático para a maior parte das pessoas. Os gráficos da pesquisa encontram-se no APÊNDICE A.

90% das pessoas responderam que utilizam ou já utilizaram as vagas de EstaR. Entre elas, 72% permanecem, em média, menos que 1 hora na vaga e apenas 10% consideraram fácil adquirir um cartão EstaR no momento necessário. 74% das pessoas responderam que já estacionaram em vaga de EstaR sem utilizar o cartão, sendo que as principais justificativas foram: permaneceria pouco tempo estacionado para realização de alguma atividade rápida e dificuldade em comprar o cartão onde estacionou. Estes dados demonstram que há um problema no sistema atual, não só pela dificuldade em se obter um cartão no momento necessário, mas também pela necessidade de permanência na vaga por pouco tempo, fazendo com que não utilizem o cartão EstaR.

57% das pessoas já foram multadas por infração em vagas de EstaR, sendo os principais motivos: não utilização do cartão EstaR, permanência na vaga além do tempo permitido e preenchimento incorreto do cartão. 66% das pessoas já disperdiçaram um cartão EstaR uma ou mais vezes devido a um preenchimento incorreto. Este é um problema que também surge no sistema atual, a forma de preenchimento do cartão.

Foi perguntado ainda, se as pessoas prefeririam pagar pelo tempo real estacionado em minutos, ao invés do sistema atual que é por horas, sendo que 91% das pessoas responderam que sim. Foi proposta ainda, a substituição do atual sistema por um aparelho eletrônico no qual os créditos poderiam ser recarregados em casa, com 1 crédito equivalente a 1 minuto estacionado e foi perguntado se as pessoas optariam pela utilização deste aparelho. 21% responderam que talvez utilizariam, dependendo do custo, e 75% responderam que sim. As justificativas pela escolha do aparelho foram, principalmente, praticidade, facilidade, comodidade e sistema mais justo, conforme nuvem de palavras (Figura 1) elaborada através do site *Wordle* [1] a partir de todas as justificativas.

Figura 1 – Pesquisa de Campo: porque as pessoas utilizariam o dispositivo eletrônico
Fonte: http://www.wordle.net/

Com o sistema de estacionamento regulamentado eletrônico, o usuário pagaria apenas pelo tempo real estacionado, uma vez que o dispositivo descontaria os créditos a cada minuto, a partir do momento estacionado. Sua utilização eliminaria os erros de preenchimento do cartão EstaR. O usuário também teria a possibilidade de realizar a recarga de créditos em seu próprio domicílio através de um computador, ao invés de ter a necessidade de procurar algum posto de venda. A necessidade de um sistema mais seguro contra falsificações, quando comparadas ao sistema de blocos de cartões de papel, também se mostrou necessária. Com o sistema eletrônico diminui-se a intensidade do comércio clandestino destes blocos e, com a comunicação criptografada entre computador e dispositivo, dificultam-se as fraudes.

1.2 OBJETIVOS

O objetivo principal deste projeto é desenvolver um sistema eletrônico de estacionamento regulamentado de fácil manuseio pelo usuário, como uma alternativa tecnológica ao sistema de blocos de cartões utilizados em Curitiba. Busca-se um sistema em que o usuário não precise pagar uma hora cheia se ele

estaciona por menos tempo, isto é, pague pelo tempo real estacionado. Dessa forma, o usuário não necessita fazer uma previsão de quanto tempo ele ficará estacionado, desde que permaneça estacionado menos do que o tempo máximo permitido.

Como objetivos específicos, pode-se citar:

1- O desenvolvimento de um dispositivo embarcado portátil que realize o desconto de créditos por minuto estacionado e preferivelmente com baixo consumo elétrico para que possa ficar ligado durante as várias horas de uso. Esse sistema embarcado também deve ser de manuseio prático, para que o usuário não perca tempo para inicializar sua função principal.

2- A construção de um *website* no qual o usuário poderá se cadastrar e efetuar a compra dos créditos para seu dispositivo embarcado.

3- Possibilitar a comunicação entre o *website* e o dispositivo eletrônico para que possa haver a transferência dos créditos comprados através da porta USB, uma vez que portas USB são comuns em qualquer computador atualmente.

4- A aplicação de algoritmos de criptografia no número de créditos dos usuários, para que o sistema se torne mais seguro e mais difícil de ser fraudado.

1.3 DIAGRAMA DO SISTEMA

A representação geral do sistema pode ser vista no diagrama de blocos da Figura 3 e uma representação mais detalhada na Figura 3. O usuário que comprar o dispositivo embarcado fará o cadastro no *website* e possuirá uma chave de criptografia vinculada à conta e àquele dispositivo embarcado. Ao fazer a compra pelo *website*, seus créditos serão criptografados e ficarão armazenados no banco de dados do *website*, até que ele faça a transferência de seus créditos conectando o dispositivo embarcado no computador pela porta USB.

Após a transferência dos créditos, o dispositivo fará a decriptografia para interpretar o número de créditos recebidos. Quando o usuário iniciar o a função principal do dispositivo, o desconto de créditos será iniciado, minuto a minuto. Na interface entre o dispositivo e o usuário existem simples botões de ajuste para que o

usuário navegue entre as opções oferecidas e o visor, onde o usuário e o fiscal poderão conferir visualmente o desconto dos créditos e o tempo em que o usuário está estacionado.

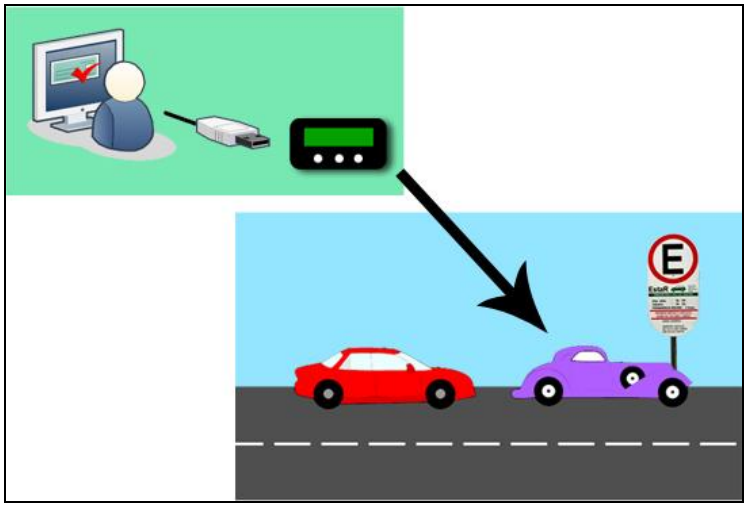


Figura 2 – Diagrama simplificado do sistema
Fonte: Autoria própria

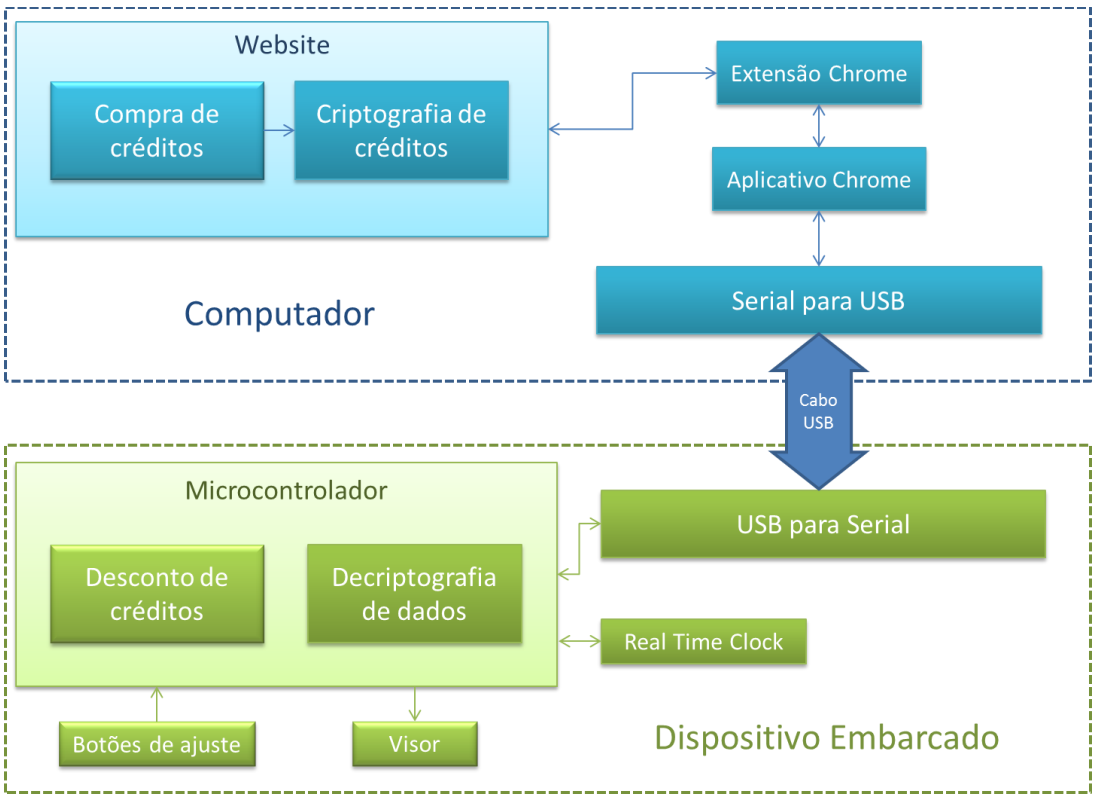


Figura 3 – Diagrama geral do sistema
Fonte: Autoria própria

1.4 ORGANIZAÇÃO DO DOCUMENTO

O primeiro capítulo deste relatório tratou da justificativa para a realização do projeto, sobre o diagrama geral de funcionamento e os objetivos principais a serem alcançados com o desenvolvimento do trabalho. O capítulo 2 descreve as tecnologias utilizadas e o motivo de suas escolhas. O capítulo 3 apresenta informações técnicas sobre o como foi desenvolvido o projeto, dividido em construção de h*ardware*, de *firmware* e de s*oftware*. No capítulo 4 é descrito o funcionamento geral do sistema, além da análise de resultados e de testes de autonomia no dispositivo embarcado. O capítulo 5 contém dados sobre a gestão do projeto, assim como o cronograma planejado e executado e análise de custos e riscos. O capítulo 6 finaliza o relatório com considerações finais sobre o desenvolvimento do projeto e as possibilidades de trabalhos futuros para aperfeiçoamento do sistema desenvolvido.

2 DESCRIÇÃO DO SISTEMA E TECNOLOGIAS ENVOLVIDAS

2.1 MICROCONTROLADOR MSP430

Para escolher o microcontrolador do dispositivo embarcado foram realizadas pesquisas para verificar o consumo elétrico, a suficiência de processamento e de módulos de comunicação que seriam necessários e o preço. Não é necessário que o dispositivo tenha velocidade de processamento alta, uma vez que, não serão desenvolvidas, por exemplo, atividades complexas com áudio e imagem que geralmente requerem *clock* por volta de 50MHz ou mais. O pior caso ocorreria caso o microcontrolador precisasse realizar o processo de decriptografia, desconto e armazenagem de créditos na memória, em no máximo um minuto, que é o tempo que terá para repetir o processo novamente. Isto não torna-se crítico, uma vez que uma base de tempo de um minuto é extremamente alta para microcontroladores, mesmo que sua velocidade de processamento seja baixa.

O MSP430G2xxx é um microcontrolador de 16 bits com consumo de energia baixo – em modo ativo utiliza 230µA quando em 1MHz e 2.2V –, arquitetura RISC, tensão de alimentação entre 1.8V e 3.6V, com configuração de *clock* até 16MHz, memória *flash* de 16kB, memória RAM de 512 Bytes e 14 pinos I/O de propósito geral [2].Sua arquitetura interna pode ser vista na Figura 4. Tais características são mais do que suficientes para o projeto em questão.

O microcontrolador, ainda, possui módulos de comunicação I2C, necessário para a comunicação do microcontrolador com o display OLED e com o Relógio de Tempo Real (*Real time clock* ou RTC) e módulo de comunicação UART, que pode ser usado para comunicação entre o dispositivo e o computador.

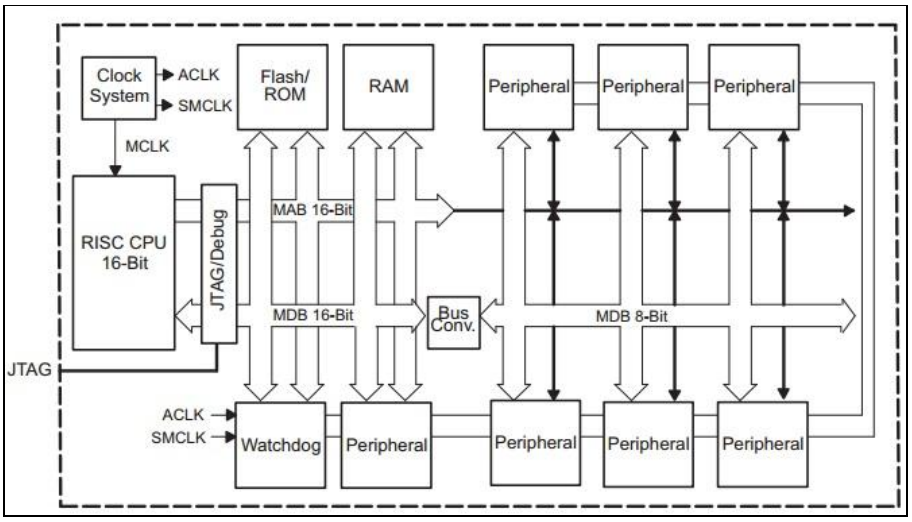


Figura 4 – Arquitetura interna do MSP430G2553
Fonte: *Datasheet* do microcontrolador [2]

2.2 *KIT* DE DESENVOLVIMENTO *LAUNCHPAD* MSP-EXP430G2

A empresa Texas Instruments produz uma linha de *kit* de desenvolvimento, o *LaunchPad* MSP-EXP430G2. Esse *kit* (Figura 5) foi desenvolvido justamente para que pudessem ser aproveitados recursos do microcontrolador MSP430G2553 facilmente – que é o quarto microcontrolador com mais recursos de um total de 49 da série *Value Line* da Texas Instruments [2] - disponibilizando toda a parte de emulação e debug na placa, inclusive a emulação da UART através da porta USB, para que seja possível utilizá-la para comunicação com o computador .[3]

Figura 5 – *Kit* de desenvolvimento *LaunchPad* MSP-EXP430G2
Fonte: Manual do *kit* [4] **[3]**

Além disso, o *kit* de desenvolvimento possui tamanho ideal para a aplicação, cerca de 5,0x6,5cm, tornando-se portátil e prático para o usuário.

As IDEs - Ambiente Integrado de Desenvolvimento (do original *Integrated Development Environment*) mais comuns e compatíveis com ele são o *IAR Embeedded Workbench* (IAR*)* e o *Code Composer Studio* (CCS) [3] . No início do desenvolvimento deste projeto, foi utilizado o IAR. Posteriormente, foi preciso migrar para o CCS pois o IAR fornece até 4kb de código na sua versão sem custos[4] . Quando foi atingido esse limite, foi necessário migrar para o CCS, que fornece até 16kb de código[5]. O código completo ficou com aproximadamente 8kb, assim não foi preciso efetuar a compra da licença de nenhuma das IDEs.

2.3 RELÓGIO DE TEMPO REAL PCF85863

Os critérios para a escolha do *Real Time Clock* (Relógio de tempo real ou RTC) a ser utilizado no projeto foram: mesma tensão de alimentação do microcontrolador, que no caso é de 1.8V a 3.6V e função de alarme ou *timer*, para que possa ser usado como base para o desconto de créditos de minuto a minuto.

O PCF8563 produzido pela empresa NXP é um RTC otimizado para aplicações de baixo consumo e fornece o ano, mês, dia, dia da semana, hora, minuto e segundo baseado em um cristal de 32.768kHz, tensão de operação entre 1V e 5.5V, saída de clock programável por s*oftware*, funções de alarme e *timer*, pino de saída de interrupção e comunicação I2C de até 400kb/s [6]. Estes parâmetros atendem aos requisitos necessários para a aplicação.

Ele possui 8 pinos, que podem ser vistos na Figura 6, sendo 2 deles para comunicação I2C, 2 para a entrada e saída do cristal oscilador, 2 para alimentação de tensão 1 pino de interrupção gerada pelo alarme ou pelo timer e 1 pino para o *clock* configurável de saída.

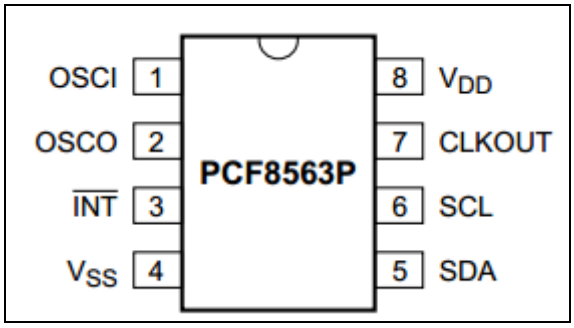


Figura 6 – Relógio de tempo real PCF8563
Fonte: *Datasheet* do fabricante [6]

2.4 VISOR

Foram pesquisados visores que já são acompanhados de seus controladores, facilitando a interface do microcontrolador com o *display*. Priorizou-se *displays* de menor custo, com tensão de alimentação de até 3.6V e, principalmente, que suportam temperaturas mais altas, visto que o interior de um carro em um dia com a temperatura ambiente de 40ºC em Curitiba (onde o máximo já registrado foi 35,5ºC [7]) pode chegar até 70ºC [8]

Por isso, foi considerada a possibilidade da utilização de um visor OLED (Diodo emissor de luz orgânico do original *Organic Light-emitting Diode)* . Visores OLED são mais finos e iluminam mais que LCDs, porque não utiliza luz de fundo e dois polarizadores [9] . São visores que consomem menos energia, possuem melhor contraste, apresentam um ângulo de visão maior (o que pode facilitar para o fiscal

de trânsito), têm maior durabilidade (10 mil horas a 46 lumens por Watt)[10] e podem operar em um intervalo de temperatura um pouco maior (até 75ºC em média)[11] quando comparado a um visor de LCD (até 60ºC em média)[12].Uma comparação dos principais parâmetros do LCD e OLED podem ser vistos na Tabela 1.

Tabela 1 – Comparação entre OLED e LCD

Parâmetro	LCD	OLED
Contraste	1000:1	1000000:1
Tempo de resposta	1ms a 8ms	Na base de microsegundos
Influências ambientais	Temperaturas baixas podem influenciar no tempo de respostas. Temperaturas altas podem influenciar no contraste	Água pode danificar materiais orgânico
Consumo de energia	Depende da tecnologia de backlight	Varia com o brilho e cor, mas menor do que do LCD
Ângulo de visão	Até 10° em todos os sentidos	Até 80° graus em todos os sentidos
Intervalo de temperatura	Até 60°C	Até 75°C

Fonte: Adaptado de OLED Association [9]

Foi escolhido o componente ER-OLEDM1602-4 (Figura 7) produzido pela EastRising. É um *display* OLED de 16 caracteres por 2 linhas acompanhado de um controlador (US2066), e tem como opções de interface com o microcontrolador: 4-bit Paralelo, 8-bit paralelo, I2C, 3-Wire Serial SPI e 4-Wire Serial SPI [13] A opção de comunicação I2C tornou-se conveniente para o projeto visto que este protocolo de comunicação já será implementado para a comunicação do microcontrolador com o RTC, sendo possível economizar até 6 pinos de I/O, número significante, uma vez que o microcontrolador possui 14 pinos I/O de propósito geral. As características do visor OLED podem ser vistas na Tabela 2.

Figura 7 – Visor OLEDM1602
Fonte: EASTRISING[13]

Tabela 2: Parâmetros do visor OLED

ER-OLEDM1602	
Parâmetro	Valor
Peso	0.025kg
Formato do Display	Caractere 16x2
Dimensão	80.00(W)x11.52(H)x4.00(T)mm
Area visual	58.22(W)x14.52(H)
Area ativa	56.22(W)x11.52(H)
Tamanho do caractere	2.97mmx5.57mm
Controlador associado	US2066
Interface	4-bit Paralelo 8-bit Paralelo I2C 3-Wire Serial SPI 4-Wire Serial SPI
Tensão de alimentação	3.3V
Temperatura de operação	-30°C até 70°C
Temperatura de armazenagem	-40°C até 85°C

Fonte: Adaptado de EASTSIRING[13]

2.5 BATERIA

A escolha da bateria em um projeto que envolva um sistema embarcado é muito importante. Deve-se buscar um equilíbrio entre a capacidade de energia que essa bateria pode fornecer e o seu tamanho físico [14]. O tamanho físico é um fator

determinante, pois é possível adquirir baterias que forneçam mais energia e por mais tempo que outras, mas por outro lado, tem seu tamanho físico muito maior, inviabilizando seu uso em determinado projeto. Portanto, para a escolha da bateria neste projeto, além das especificações elétricas necessárias a serem atendidas, levou-se em conta seu tamanho físico.

Foi utilizada a BL-5C (Figura 8), uma bateria recarregável do tipo Li-on de celulares da marca Nokia, mas também muito comum em aplicações embarcadas portáteis pelo equilíbrio entre seu tamanho e autonomia. Medindo 5,3cm de comprimento, 3,3cm de largura e 0,5cm de espessura, com tensão de 3.7V (sendo compatível com os circuitos do projeto), tensão de recarga de 4.2V, capacidade de 1200mAh e pesando apenas 22g.

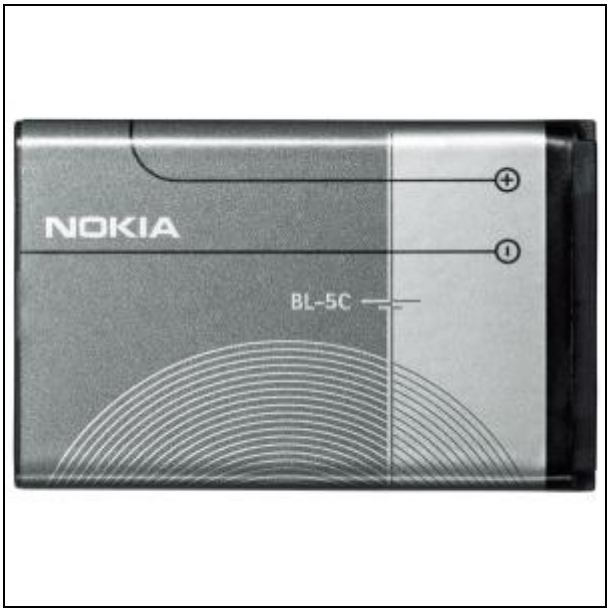


Figura 8 – Bateria BL 5C
Fonte: <www.wpsantennas.com>

2.6 MÓDULO DE RECARGA

Foi escolhido o módulo de recarga que utiliza o CI TP4056 (Figura 9) para o projeto. Por ser em encapsulamento SOP, tem tamanho reduzido, sendo assim conveniente para projetos portáteis [15]. Além disso, ele funciona com adaptadores USB, encaixando-se ao projeto em questão, visto que é muito comum o uso de

adaptadores USB para recargas de dispositivos dentro de automóveis. O TP4056 automaticamente encerra o ciclo de recarga quando a corrente de recarga atinge um décimo de seu valor máximo. Sua tensão de recarga é de 4.0V a 8.0V o que é compatível com a tensão da recarga da bateria, que é de 4.2V.

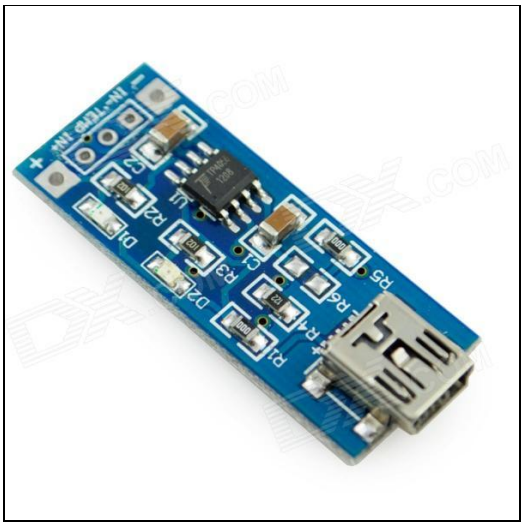


Figura 9 – Módulo de recarga CI TP4056
Fonte: Fabricante do CI TP4056 [15]

2.7 PROTOCOLO DE COMUNICAÇÃO I2C

O protocolo I2C é utilizado para a comunicação do MSP430G2553 com os periféricos, com o RTC PCF8563 e com o controlador do visor OLED US2066. O I2C é um barramento de 2 fios em que não é necessário um fio de Chip Select, para que o Master selecione o periférico com o qual trocará mensagens [16]. Os dois fios que compõe o barramento do protocolo I2C são o *Serial Data* ou SDA e o *Serial Clock* ou SCL. O SCL fornece o *clock* para que haja a sincronização entre os dispositivos e o SDA é o fio onde as mensagens serão enviadas ou recebidas. Juntos, possibilitam suporte de transmissão de *bytes* (8 bits) para dispositivos com endereços de 7 bits e um bit de controle. Uma ilustração da configuração da alocação dos dispositivos na comunicação I2C pode ser vista na Figura 10.

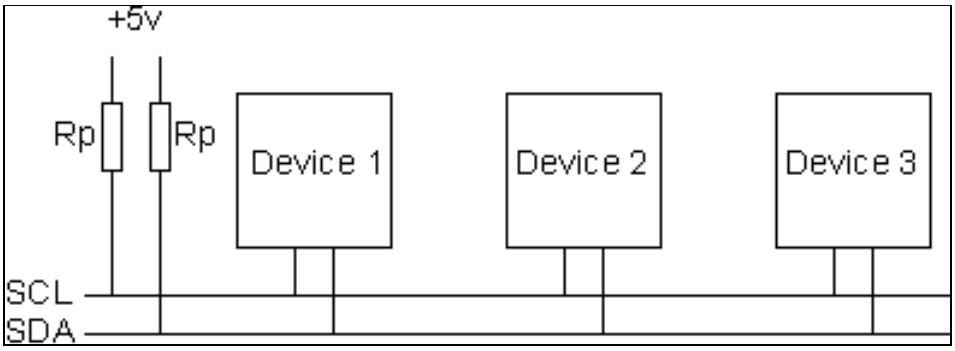


Figura 10 – Barramento I2C
Fonte: Robot Eletronics [16]

Os dispositivos conectados ao barramento I2C são *Master* (Mestre) ou *Slaves* (Escravos). *Master* são aqueles que controlam a linha de *clock*, o SCL. *Slaves* são os que respondem ao *Master*. *Slaves* não conseguem iniciar uma transferência no barramento I2C, somente os *Masters*. Uma das grandes vantagens do protocolo I2C é que se pode ter diversos *Slaves* em um só barramento I2C [16]. No caso do projeto em questão o MSP430G2553 é o *Master*, que inicializa a comunicação no momento em que achar necessário com o primeiro *Slave* PCF8563 e com o segundo Slave o visor OLED US2066. É preciso desenvolver um código nas rotinas de execução do MSP430G2553 para que ele inicialize a comunicação I2C com o PCF8563 toda vez que precisar ler ou escrever alguma informação do horário, e precisará iniciar a comunicação I2C com o US2066 toda vez que precisar escrever algo no visor OLED.

A associação de múltiplos *Slaves* no barramento é possível porque cada *Slave* possui um endereço de 7 bits. Depois do *Start*, o *Master* “joga” no barramento esse endereço mais um bit de controle *Read* ou *Write*, e, quem o responder, identificou esse endereço e enviou uma resposta para o *Master*, sinalizando o reconhecimento (ACK) [16]. A sequência de comandos do protocolo I2C pode ser vista na Figura 11.

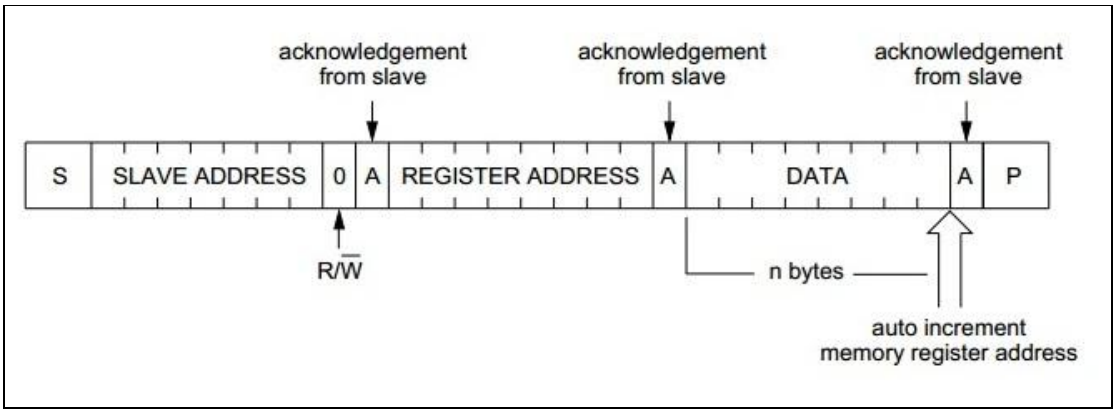


Figura 11 – Sequência de comandos do protocolo I2C
Fonte: Robot Eletronics [16]

Após a detecção do reconhecimento (ACK), o *Master* envia o endereço dentro do *Slave* onde ele deseja escrever ou ler e assim que obter outro ACK iniciará a troca de mensagens até que envie a sinalização de *Stop* [16].

Tanto o PCF8563 quanto o US2066 são limitados em um SCL de até 400kHz [6] [13] , mas já existem dispositivos que suportam até 3.4MHz [16].

O microcontrolador MSP430 oferece um módulo especifico para a utilização do protocolo I2C, no qual o programador não precisa desenvolver a sequência de sinais de início, parada e outros sinais de controle. Este assunto será abordado na seção 3.2.1.

2.8 COMUNICAÇÃO SERIAL ASSÍNCRONA COM O PC

O *kit* de desenvolvimento *LaunchPad*, oferece em sua placa o tunelamento da comunicação serial assíncrona por USB através de um chip TUSB3410 embutido. O TUSB3410 é um controlador de porta USB para serial e sua arquitetura pode ser vista no Anexo A [17].

Isto significa que é possível utilizar a comunicação serial assíncrona através da interface UART entre o microcontrolador e o computador pelo conector USB, não necessitando que o usuário precise comprar um conversor Serial/USB visto que os computadores hoje em dia não possuem mais entrada RS232 para a comunicação

serial. Uma representação ilustrativa do fluxo de informações da comunicação serial assíncrona pode ser vista na Figura 12.

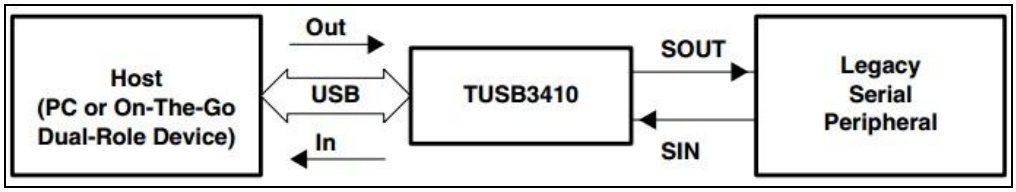


Figura 12 – Fluxo de dados com conversão USB<>Serial
Fonte: Texas Instruments [17]

Transmissão assíncrona significa que aquele que enviará uma mensagem não necessita mandar um sinal de *clock* para aquele que receberá [18]. Mas antes, transmissor e receptor devem ser configurados com os mesmos parâmetros de frequência e composição da mensagem. Quando uma palavra é dada para a interface UART, um bit de Start é adicionado no começo daquela palavra. O bit de Start tem a finalidade de alertar o receptor que uma mensagem será enviada a ele, e forçar a sincronização do *clock* do receptor com a do transmissor.

Depois do bit de *Start*, toda mensagem é enviada, iniciando pelo bit menos significativo. Ainda existe a opção de se adicionar um bit de paridade para o receptor poder verificar se houve algum erro na transmissão. Depois de recebida a mensagem, um bit de parada é mandado do transmissor para o receptor, sinalizando que finalizará a transmissão (Figura 13) [18].

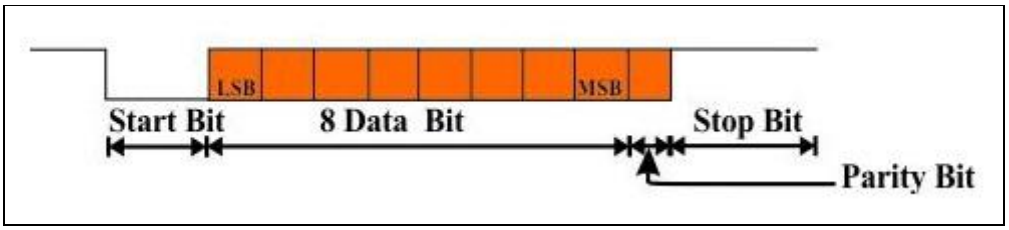


Figura 13 – Sequência de comandos transmissão serial
Fonte: FRANK, Durda, 2013 [18]

Há várias frequências de operação para a interface UART. Basta que receptor e transmissor estejam configurados com a mesma frequência. Porém, com o cristal utilizado no *kit LaunchPad* MSP-EXP430G2 a interface UART do MSP430

tem um limite de 9.600b/s [3] . Então é essa frequência que é configurada no computador para a transmissão serial.

2.9 ALGORITMOS DE CRIPTOGRAFIA

A troca de informações sigilosas é uma necessidade recorrente. A criptografia é uma ferramenta que utiliza um conjunto de regras que visa codificar a informação de modo que apenas o emissor e receptor consigam decifrá-la. A idéia é transformar um texto compreensível em um texto codificado usando um algoritmo matemático [19].

Um algoritmo de criptografia acompanha uma chave que é usada para criptografar, ou seja, transformar um texto compreensível em um texto codificado, e uma para realizar o processo inverso, para decriptografar. O tipo ou tamanho das chaves a serem utilizadas dependem do algoritmo de criptografia e o quanto de segurança é necessário naquele sistema [20].

Na criptografia simétrica (Figura 14) uma única chave é usada. Com essa chave, o transmissor pode criptografar a mensagem e o receptor pode decriptografar com essa mesma chave. Os algoritmos que são usados para a criptografia simétrica são mais simples do que os algoritmos usados na criptografia assimétrica. Em função desses algoritmos serem mais simples, e da mesma chave ser usada para criptografar e decriptografar dados, a criptografia simétrica é muito mais rápida que a assimétrica [21]. Uma das principais desvantagens da criptografia simétrica é que todas as partes que enviam e recebem os dados devem conhecer ou ter acesso à chave de criptografia. Esse requisito cria um problema de gerenciamento de segurança e problemas de gerenciamento de chave que uma organização deve considerar em seu ambiente. Este problema de gerenciamento de segurança existe porque a organização deve enviar a chave de criptografia a todos que requisitarem acesso aos dados criptografados. Os algoritmos de criptografia simétrica utilizados mais comuns são: DES (*Data Encryption Standard*), 3DES (*Triple Data Encryption Standard*) e o AES(*Advanced Encryption Standard*) [21].

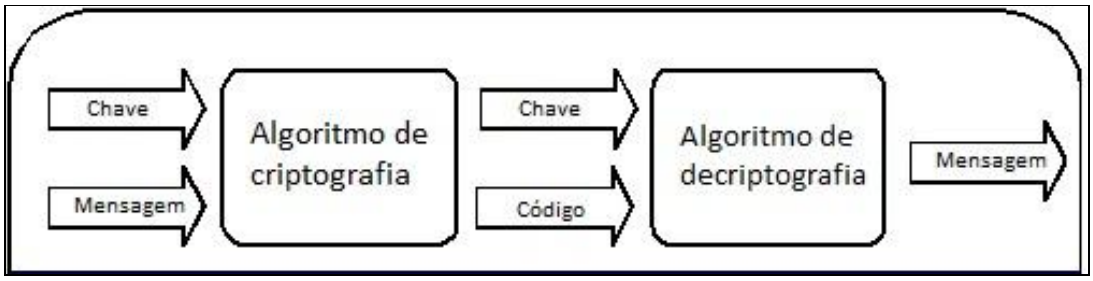


Figura 14 – Processo de criptografia/decriptografia simétrica
Fonte: Adaptado de NETWORK SORCERY[20]

Na criptografia assimétrica(Figura 15) a chave para criptografar é diferente da utilizada para decriptografar. A chave utilizada pelo transmissor para criptografar é pública e a utilizada para decriptografar é uma chave privada que apenas aquele receptor possui, mas que está matematicamente ligada à chave pública. Este método é considerado mais seguro que a criptografia simétrica [21]. Contudo, como a criptografia assimétrica usa algoritmos mais complexos do que a simétrica, o processo de criptografia é muito mais lento e requer muito mais custo computacional. A criptografia assimétrica é muito mais usada em sistemas em que há vários transmissores. Assim, esses transmissores podem usar a mesma chave pública para criptografar, e sabem que apenas aquele receptor, que possui aquela chave privada, poderá decriptografar. Os algoritmos assimétricos mais comuns são o RSA (*Rivest-Shamis-Adleman*) e o DSA (*Digital Signature Algorithm*) [21].

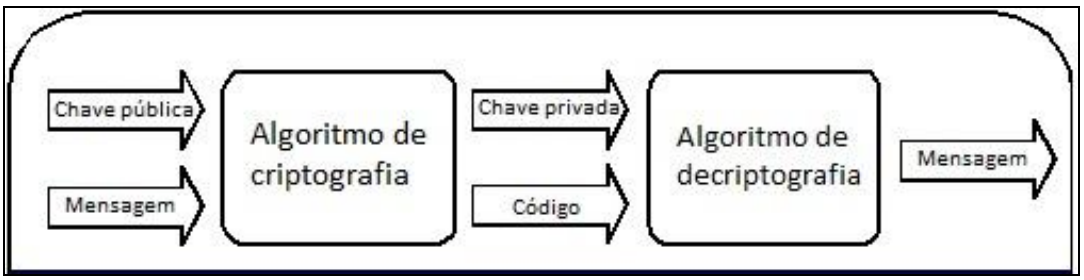


Figura 15 – Processo de criptografia/decriptografia assimétrica
Fonte:Adaptado de NETWORK SORCERY [20]

Neste projeto escolheu-se a utilização de criptografia simétrica pelo fato de que é um algoritmo muito mais simples e se tornando eficaz para nossa aplicação. O maior problema da criptografia simétrica é o fato de que tanto transmissor quanto receptor precisam ter acesso a chaves iguais. Isso pode ser contornado no processo de fabricação e de configuração inicial do dispositivo. Não é necessário enviar a

chave junto com a mensagem. Transmissor e receptor terão a chave previamente, sem risco de intercepção e alteração da chave. O método de implementação será detalhado no capítulo **Error! Reference source not found.**. Assim, é construído um sistema seguro, sem a necessidade de gastar recursos computacionais desnecessários. Nota-se a complexidade do algoritmo assimétrico ao comparar-se, por exemplo, a necessidade de criar uma chave de 2048 bits no algoritmo RSA para que tenha a segurança equivalente a uma chave de 256 bits do algoritmo simétrico AES[20].

Entre os algoritmos de criptografias simétricos, foi escolhido o que mais se aplica ao projeto em questão. Todos eles tem suas vantagens e desvantagens. Dentre os mais utilizados, como já mencionado no começo desta seção, destacam-se o DES, 3DES e AES.

2.9.1 DES e 3DES

O DES foi desenvolvido em 1970 pela IBM e foi adotado como padrão pela NIST em 1976. O algoritmo DES foi declarado como inseguro pela NIST, enquanto o 3DES como razoavelmente seguro. O DES tem sua chave muito curta para que seja seguro e pode ser quebrado por força bruta pela máquina “EFF DES cracker” construída em 1998 [22].

O 3DES é um truque de reutilização do DES em que se cascateiam três blocos de criptografia DES. É considerado razoavelmente seguro, mas é bastante lento, especialmente em *software*, uma vez que foi construído para implementações em *hardware* [22].

A estrutura do algoritmo de criptografia DES pode ser vista no Anexo B.

2.9.2 A Criptografia AES

O AES foi anunciado pelo NIST em Novembro de 2001. É o sucessor do DES, que não poderia mais ser considerado seguro por sua chave de somente 56 bits [23].

Existem três diferentes versões do AES. Todas elas tem um bloco de informação de extensão 128 bits, onde a chave pode ser de 128, 192 ou 256 bits. O AES tem um bom funcionamento tanto em *software* quanto em *hardware* [22].

O diagrama de funcionamento do AES pode ser visto no Anexo C.

Por ser um algoritmo de criptografia mais novo e mais eficiente, ele foi escolhido como o algoritmo de criptografia deste projeto. O AES com chave de 128 bits tornou-se conveniente de ser utilizado pois a empresa Texas Instruments, fabricante do microcontrolador escolhido, disponibiliza uma biblioteca gratuitamente com a implementação do AES 128. Como o objetivo do projeto é o desenvolvimento do sistema de EstaR Eletrônico e não o desenvolvimento de algoritmos complexos de criptografia, optou-se pela utilização dessa biblioteca da TI. Para um melhor entendimento do funcionamento do algoritmo, consultar documento de implementação do AES 128 [23].

Com o uso da biblioteca o processo de criptografia e decriptografia utilizam 80 bytes de memória RAM cada um, 1839 *bytes* da memória *Flash* para criptografar e 2423 *bytes* da memória *Flash* para decriptografar [23]. Foi visto no capítulo 2.1 que o MSP430G2553 possui 512 *bytes* de memória RAM e 16kB de memória *Flash*, tornando-se, portanto, um microcontrolador apto para a utilização da biblioteca disponibilizada pela fabricante.

2.10 *WEBSITE*

Para o desenvolvimento do *website* optou-se primeiramente pela utilização de modelos de lojas virtuais já desenvolvidos com possibilidade de incrementação para a venda de créditos.

O *website* a ser desenvolvido, porém, não seguiria o padrão de um *website* comum. Seria necessário um banco de dados, que associasse cada usuário com cada chave de criptografia específica daquele dispositivo embarcado que realizou a compra. Os créditos comprados precisam ser criptografados e, quando o usuário optar, estes créditos devem ser transferidos do banco de dados do site para o dispositivo embarcado.

Com todas essas necessidades adicionais para o *website*, a utilização de um modelo geral pronto não tornou-se a melhor escolha. Ainda, estes modelos de *website*s, em geral, acompanham muitos recursos e, a maior parte deles, não seriam utilizados na aplicação do projeto.

Então, optou-se pelo desenvolvimento de um *website* próprio, sem a utilização de modelos prontos. O site de hospedagem “Hostinger” já inclui em seus serviços a base de dados MySQL, suporte para PHP, HTML5 e Apache em seu pacote sem custos, o que se tornou uma boa opção para o desenvolvimento do projeto. Porém, optou-se pela hospedagem não gratuita pelo site "Zooming" que inclui os mesmo serviços necessários, uma vez que o Hostinger ficou inativo por alguns dias, prejudicando o desenvolvimento do projeto.

2.11 EXTENSÃO E APLICATIVO

A comunicação entre o dispositivo e o *website* foi intermediada por um aplicativo e uma extensão, utilizando a recente API *Serial* do Google Chrome disponibilizada em Junho de 2013 para o reconhecimento e a troca de mensagens com a porta serial [24]. Os aplicativos e extensões Chrome, enquanto em desenvolvimento, podem ser testados carregando-os em "Modo do desenvolvedor" no navegador através do endereço "chrome://extensions" (Figura 16).

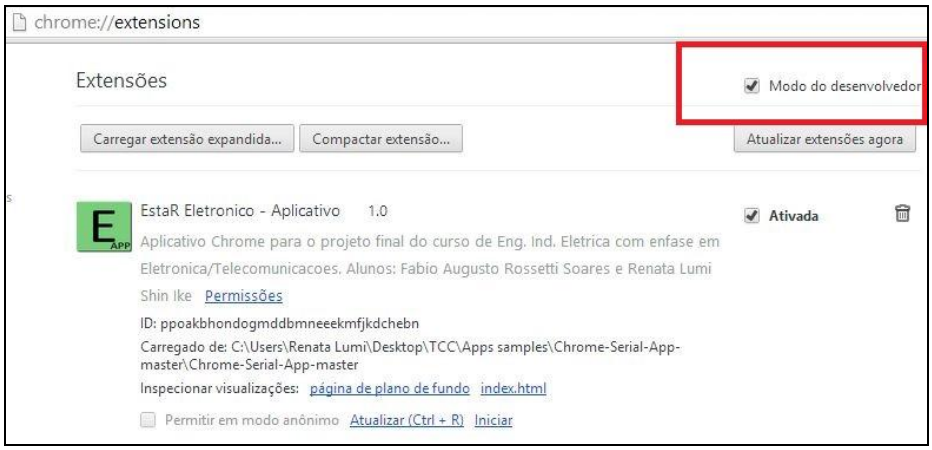


Figura 16 – Habilitando o modo desenvolver no Google Chrome
Fonte: Autoria própria

As extensões são programas que, ao serem adicionadas ao navegador, podem melhorar e/ou modificar sua funcionalidade. Por serem diretamente ligadas a ele, elas também são fortemente ligadas a web, podendo então realizar a comunicação com o *website* do projeto. Porém, como será detalhado no capítulo 3.4, ao manter a conexão web, a extensão não pode comunicar-se com a serial, sendo necessário o desenvolvimento do aplicativo Chrome.

Aplicativos Chrome têm a característica de não serem dependentes do navegador, podendo interagir não só com dispositivos de rede, mas também com dispositivos de *hardware* [25]. Eles são carregados localmente e operam principalmente *offline*, o que foi essencial para o projeto, pois pode-se utilizar a API *Serial* no aplicativo.

Assim, para a comunicação do dispositivo com o *website*, foi utilizado o aplicativo Chrome para identificar a porta serial, e a extensão Chrome para comunicação com a *web*. E, entre eles, foi utilizada a API *messaging* do Google Chrome que possibilita a troca de mensagens entre extensões/aplicativos. A Figura 17 ilustra como foi feita a comunicação.

Figura 17 – Comunicação entre *website* e porta serial
Fonte: Autoria própria

3 DESENVOLVIMENTO

Após a pesquisa de todos os componentes e de todas as tecnologias envolvidas iniciou-se o desenvolvimento do projeto. Antes do desenvolvimento do código de programação do microcontrolador e sua interface com os periféricos, foi projetado o *hardware*.

3.1 *HARDWARE*

3.1.1 Visor

O primeiro periférico a ser integrado foi o visor OLED. O visor OLED possui 15 pinos, sendo sua configuração original conforme Tabela 3.

Tabela 3 : Pinos do módulo do visor ER- OLEDM1602

Definição de pinos ER-OLEDM1602			
Pino	Símbolo	Tipo	Função
1	VSS	Alimentação	GND
2	VDD	Alimentação	Entrada 3.3V de alimentação
3 a 10	D7 a D0	I/O	Barramento de informação bidirecional de oito bits.
11	RD	Entrada	Sinal de leitura
12	RW	Entrada	Sinal de escrita
13	DC	Entrada	Pino de controle de informação/comando.
14	RSE	Entrada	Pino de reset (Ativo em baixo)
15	CS	Entrada	Chip select (Ativo em baixo)

Fonte: Adaptado de EASTRISING [13]

Como apresentado anteriormente no capítulo 2.4, a comunicação com o microcontrolador é a I2C, pois este protocolo também será usado na comunicação do controlador com o RTC, economizando pinos de I/O.

Para que o visor ER-OLEDM1602 possa ser utilizado no protocolo I2C, as configurações dos pinos foram alteradas. Os pinos D2 e D1 foram conectados juntos sendo, então, a linha SDA e a linha SCL foi designada ao pino D0. Os pinos de sinal de leitura, escrita e *chip select* foram conectados ao GND, pois são desnecessários no I2C, uma vez que o *Master* enviará no SDA o endereço do S*lave* junto com o bit de controle de leitura ou escrita. O pino de informação ou comando também foi conectado ao GND, pois também será incluso na sequência de troca de mensagens do I2C. O pino de RSE foi conectado normalmente ao microcontrolador pois é ele quem envia o sinal para que o display seja ligado. Dessa forma, apenas três pinos de I/O do microcontrolador serão utilizados.

O esquemático da Figura 18 mostra como o display foi conectado para a utilização do protocolo I2C.

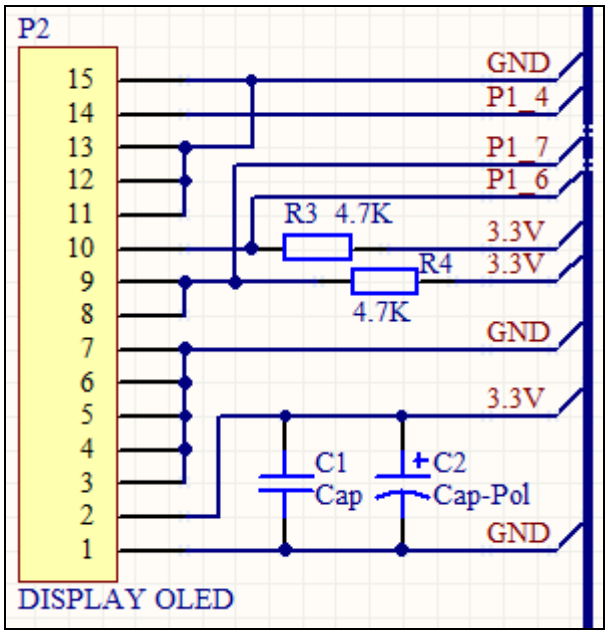


Figura 18 – Conexão do visor OLED
Fonte: Autoria própria

O barramento I2C exige a adição de resistores de *pull-up* nos pinos de SDA e SCL. O valor destes resistores podem ser por volta de 5k Ohms. Neste projeto utilizaram-se resistores de 4.7k Ohms.

A versão inicial em *protoboard* (placa de ensaio) pode ser vista na Figura 19.

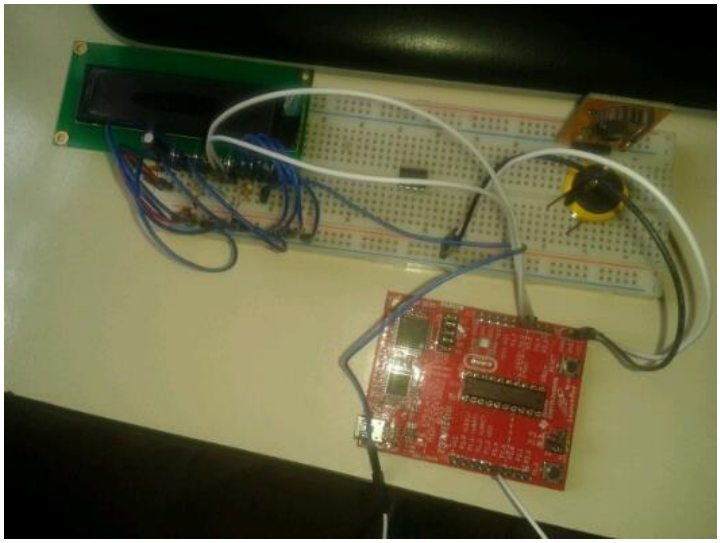


Figura 19 – Protótipo inicial - conexões do visor OLED
Fonte: Autoria própria

Uma dificuldade encontrada foi que, mesmo com todos os pinos conectados corretamente conforme o *datasheet*, e com a implementação da comunicação I2C, o visor OLED não funcionou. Assim, o fornecedor do *display* foi contatado, uma vez que poderia ser um defeito no componente. Em resposta, ele alertou a necessidade de alteração das posições dos três resistores de 0 Ohms SMD contidos na parte traseira da placa do visor OLED. Eles exercem a função de três pinos de configuração de interface. A combinação dos resistores que determina se o controlador do visor OLED US2066 se comunicará em paralelo 4 bit, paralelo 8 bit, serial ou I2C. Como os componentes não eram jumpers, a forma de escolha da interface desejada não foi facilmente identificada.

As configurações de interface podem ser vistas na Tabela 4.

Tabela 4: Combinações de jumpers associados ao modo de operação do visor OLED

Configuração da interface US2066			
BS2	BS1	BS0	Interface
0	0	0	Serial
0	0	1	Invalid
0	1	0	I2C
0	1	1	Invalid
1	0	0	8-bit 6800 paralelo
1	0	1	4-bit 6800 paralelo
1	1	0	8-bit 8080 paralelo
1	1	1	4-bit 8080 paralelo

Fonte: Adaptado de EASTSIRING [13]

A placa, quando comprada, vem configurada por padrão com 8 bits 8080 paralelo. Assim, o BS2 foi alterado para que sua entrada fosse nível de sinal baixo, alterando a comunicação para I2C. Após essa modificação, o visor funcionou conforme o esperado.

3.1.2 Relógio de tempo real

Com o funcionamento correto do visor OLED, o RTC foi integrado. O PCF8563 possui oito pinos, conforme Tabela 5.

Tabela 5: Descrição dos pinos do RTC PCF8563

PCF8563		
Símbolo	Pino	Descrição
OSCI	1	Entrada do oscilador
OSCO	2	Saída do oscilador
INT	3	Saída da interrupção
Vss	4	GND
SDA	5	Serial Data Input e Output
SCL	6	Serial Clock Input
CLKOUT	7	Saida do Clock
Vdd	8	Alimentação

Fonte: NXP SEMICONDUCTORS [6]

No esquemático da Figura 20 podem ser vistas as conexões do RTC.

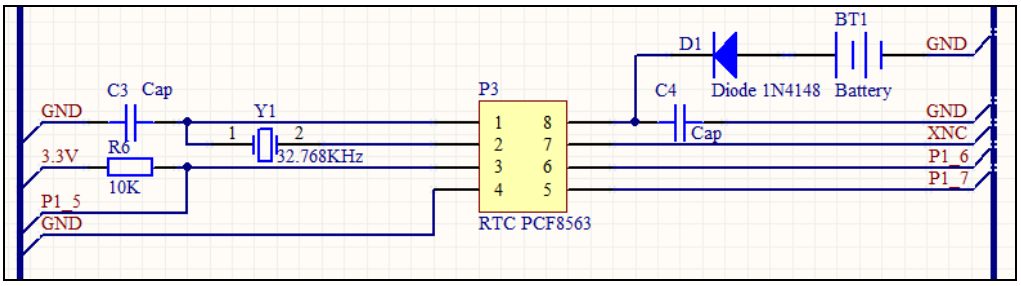


Figura 20 – Esquemático de conexões do RTC PCF8563
Fonte: Autoria própria

O freqüência do cristal oscilador utilizado é 32.768kHz. Como o RTC precisa funcionar em tempo integral, isto é, continuar a contar o tempo mesmo que a bateria do dispositivo seja retirada, ele foi alimentado por uma outra bateria. Esta bateria é uma CR2032 (Figura 21) e é amplamente utilizada em aplicações que envolvam um RTC e previne que o relógio perca suas configurações assim que se esgotar a bateria principal.

Figura 21 – Bateria CR2032
Fonte: Autoria própria

Os pinos seis e sete, SDA e SCL respectivamente, foram conectados ao mesmo barramento I2C utilizado no *display* e os dois resistores de *pull-up* também foram necessários. No esquemático da Figura 20 não são mostrados os resistores, pois eles já estão conectados no barramento I2C, visto circuito do visor OLED (Figura 18). A interrupção ocorre no pino três, que é um pino de dreno aberto e ativo em nível de sinal baixo, sendo necessário um resistor de *pull-up*. Neste circuito, seu valor é de 10k Ohms.

3.1.3 Módulo de recarga

O CI TP4056 possui oito pinos, que podem ser vistos na Tabela 6.

Tabela 6: Descrição dos pinos do módulo de recarga TP4056

TP4056		
Símbolo	Pino	Descrição
TEMP	1	Conectando o pino TEMP a um NTC conectado à bateria é possível realizar medição da temperatura no momento da recarga, e caso a temperatura registrada for muito alta ou muito baixa por mais de 0,15 segundos quando comparado com o padrão a recarga é automaticamente suspensa.
PROG	2	A corrente de recarga pode ser programada conectando-se um resistor adequado (Rprog) neste pino. A corrente de recarga será 1200 / Rprog em Amperes.
GND	3	-
Vcc	4	-
BAT	5	Conecta-se no terminal positivo da bateria a ser carregada.
STDBY/	6	Quando a bateria encerra o processo de recarga, o pino STDBY/ é direcionado ao nível baixa de tensão.
CHRG/	7	Quando a bateria está sendo recarragada, o pino CHRG/ é direcionado ao nível de baixa tensão.
CE	8	Ao nível alto de tensão, irá habilitar todas as funcionalidades do chip. Ao nível baixo, o TP4056 estará desabilitado.

Fonte: Adaptado de NANINJING TOP POWER ASIC [15]

Entretanto, o módulo de recarga comprado já acompanha todo o circuito necessário para o ciclo de recarga, basta conectar a bateria nos terminais corretos e a entrada da fonte de alimentação, que é adaptada a um conector *mini*-USB. Neste módulo o resistor R_{prog}, responsável pela quantificação da corrente de recarga, é de 1.2k Ohms, tornando a corrente de recarga igual a 1000mA.

Como foi utilizado o MSP430G2553 juntamente com a parte de emulação do *kit LaunchPad* devido a já existência do conversor USB/Serial como visto no capítulo 2.8, foi necessário criar uma adaptação para que o dispositivo não tivesse duas entradas mini-USB, sendo uma para recarga da bateria e outro para a comunicação assíncrona com o computador. Portanto, foram conectados os fios responsáveis pela parte da comunicação da entrada mini-USB do módulo de recarga na entrada mini-USB do *LaunchPad*, garantindo a comunicação assíncrona com o PC e ao mesmo tempo a recarga da bateria. Ainda nesse circuito, foi adicionada uma chave DPDT para que pudesse ser feito a escolha da utilização da bateria para alimentação do dispositivo embarcado, ou a utilização direta da fonte de alimentação mini-USB. Caso não esteja conectada a fonte de alimentação mini-USB, o dispositivo embarcado é desligado.

3.1.4 Protótipo inicial

Após os testes dos circuitos separadamente, foi implementada uma versão do *hardware* em uma placa universal, para que riscos de mau-contato causados pelo circuito em *protoboard* fossem eliminados, aumentando, assim, a miniaturização do protótipo. O quadro resumo de como ficaram as conexões do MSP430 é apresentado na Tabela 7.

Tabela 7: Conexões do microcontrolador MSP430 no projeto em questão.

MSP430		
Pino	Símbolo	Função
1	VCC	Alimentação 3.3V
2	P1.0	NC
3	P1.1	UART TX
4	P1.2	UART RX
5	P1.3	NC
6	P1.4	Reset OLED
7	P1.5	Entrada da interrupção do RTC
8	P2.0	NC
9	P2.1	Chave táctil 1 - Navegação
10	P2.2	Chave táctil 2 - Selecionar
11	P2.3	Chave táctil 2 - Voltar
12	P2.4	NC
13	P2.5	NC
14	P1.6	SCL
15	P1.7	SDA
16	RST	NC
17	TEST	NC
18	XOUT	NC
19	XIN	NC
20	GND	GND

Fonte: Autoria própria

Os pinos P1.1 e P1.2 já são conectados internamente no *kit* de desenvolvimento *LaunchPad* pois a UART é simulada pela USB.

As chaves tácteis são *push buttons* e foram necessárias três. Um botão para navegação entre menus, outro para selecionar o que estiver mostrando no visor e outro para voltar. Mais detalhes serão discutidos no capitulo **Error! Reference source not found.**. O circuito dos botões segue na Figura 22 e o protótipo inicial pode ser visto na Figura 23.

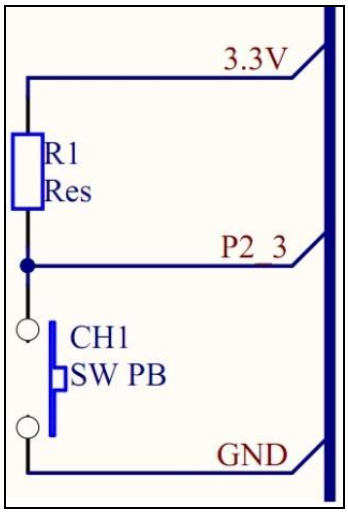


Figura 22 – Conexões das chaves tácteis
Fonte:Autoria própria

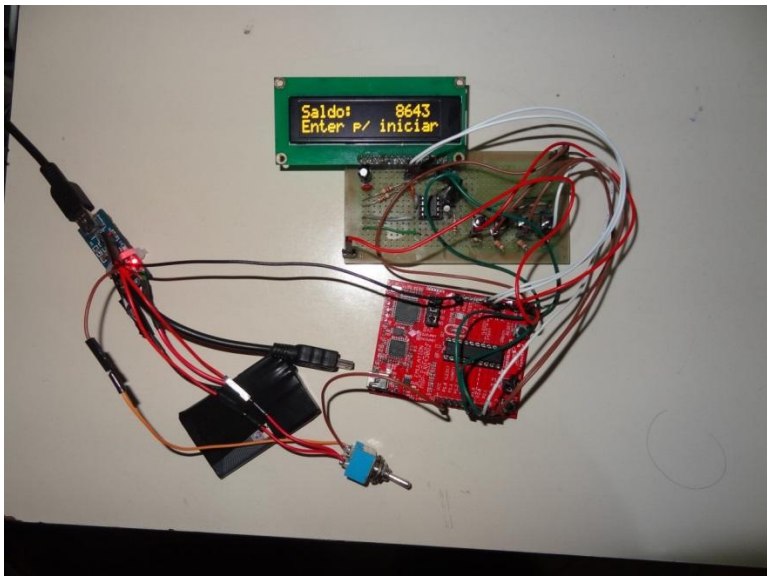


Figura 23 – Foto do protótipo inicial
Fonte: Autoria própria

3.1.5 Simulação de um protótipo final

Após a realização dos testes necessários, chegou-se a uma versão final do circuito do projeto e foi simulada a montagem de um protótipo final.

Foi elaborado o *layout* da placa de circuito impresso utilizando o *software* Altium Designer. No *layout* (Apêndice C) foram consideradas trilhas largas de 40 mils (1.016mm) com o intuito da corrosão ser caseira em uma placa simples face. Como seriam 3 módulos principais (*display*, *launchpad* e placa de circuito impresso principal), fez-se uma simulação mecânica 3D para posicioná-los de forma que o protótipo ficasse na forma mais compacta possível. A vista frontal e inferior vista na Figura 24 e o posicionamento do display e do *kit launchpad* podem ser vistos na Figura 25 e na Figura 26.

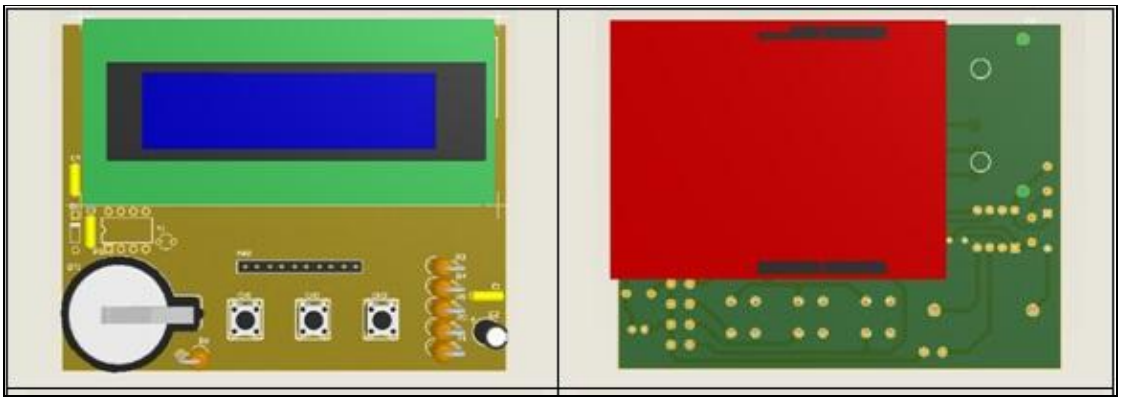


Figura 24 – Vista frontal e inferior - simulação 3D
Fonte: Autoria própria

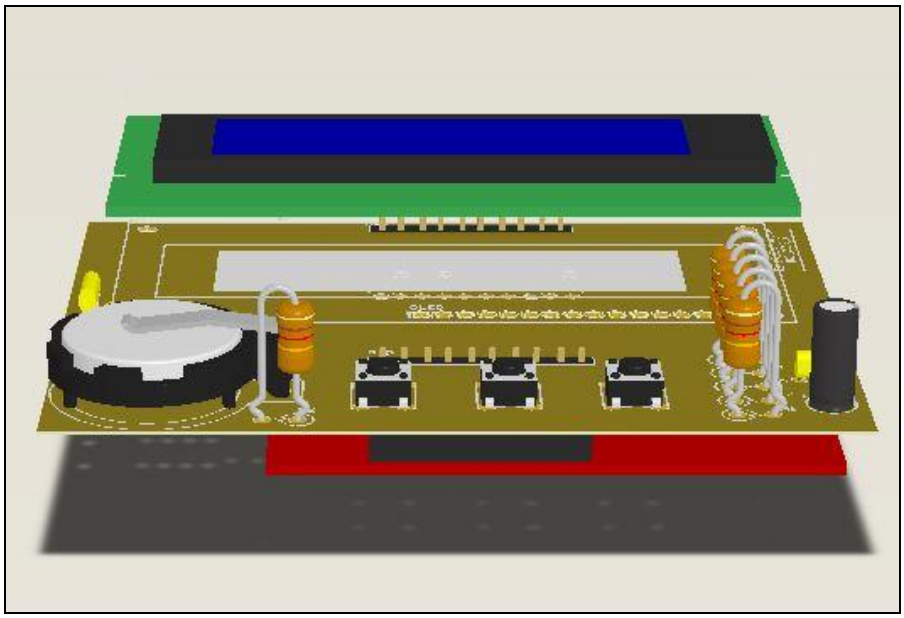


Figura 25 – Posicionamento do display - simulação 3D
Fonte: Autoria própria

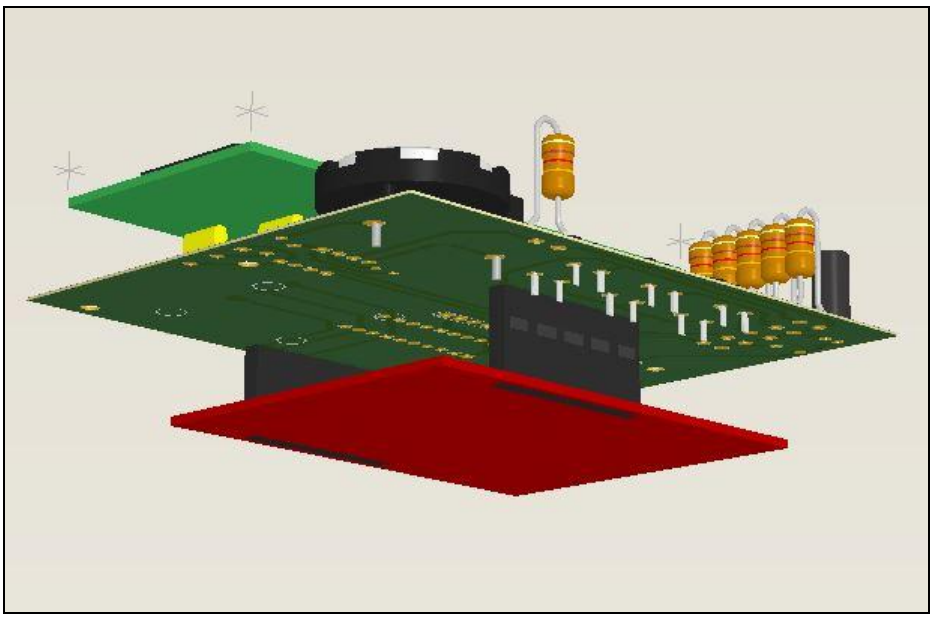


Figura 26 – Posicionamento do *launchpad* - simulação 3D
Fonte: Autoria própria

3.2 *FIRMWARE*

Foi programada uma máquina de estados para conectar todas as funcionalidades do dispositivo ao menu e facilitar a interação entre o usuário e o

dispositivo embarcado. Essa máquina de estados foi desenvolvida com o auxílio de uma *macro* para *Microsoft Excel* disponibilizada pela Texas Instruments [26]. Nessa *macro,* é necessário apenas preencher os estados e as transições correspondentes para cada evento. Feito isso, é gerado código em C específico para o MSP430G2553 e cabe ao programador desenvolver as funções de transições desejadas. A máquina de estados desenvolvida pode ser vista na Figura 27.

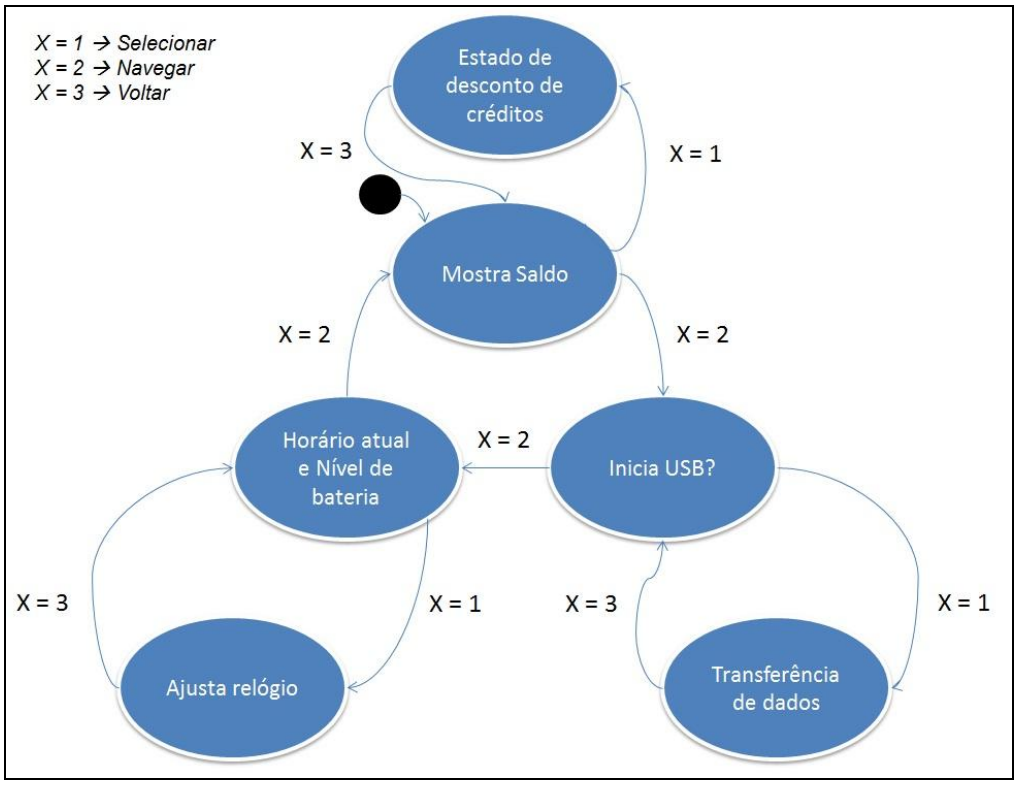


Figura 27 – Máquina de estados desenvolvida para seleção de opções
Fonte: Autoria própria

Ao ligar o dispositivo, após a tela de apresentação, o estado inicial aparecerá. Nesse estado, o usuário poderá ver a quantidade de créditos armazenada no dispositivo. Caso o usuário aperte a tecla “Selecionar”, o ele poderá escolher o tempo máximo de estacionamento naquela vaga (uma duas ou três horas) e, após selecionado, o desconto de créditos iniciará. O microcontrolador lê o horário no momento do início de descontos do RTC e imprime na tela. O processo de desconto de créditos pode ser visto na Figura 28.

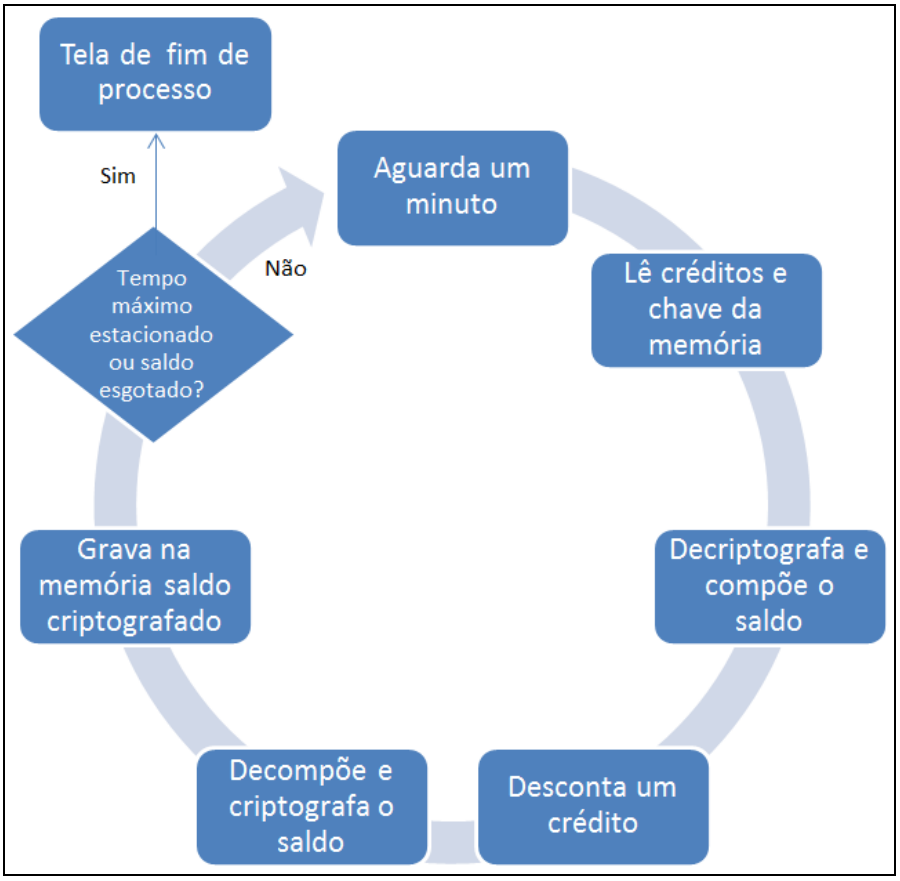


Figura 28 – Processo de desconto de créditos
Fonte: Autoria própria

A cada minuto estacionado, o saldo e a chave são recuperados da memória e ocorre o processo de decriptografia seguido pelo desconto de um crédito, para então serem impressos na tela a quantidade de minutos estacionados. Em seguida, o saldo é criptografado e gravado na memória novamente. Isso ocorre até que o usuário aperte o botão “voltar”, retornando ao menu em que é mostrado o saldo total. Se nesse momento o botão de “navegação” for pressionado, aparecerá uma mensagem “Inicia USB?”. Esse estado deve ser selecionado quando o usuário conectar o dispositivo pela porta USB no computador e desejar fazer a transferência de créditos através do *website*. Se selecionado, entrará no estado de transferência e todos os botões perderão sua função, até que o computador libere acesso novamente ao dispositivo caso a transferência seja realizada com sucesso, retornando ao menu com a pergunta “Inicia USB?”. O estado de transferência de dados pode ser visto com mais detalhes na Figura 29.

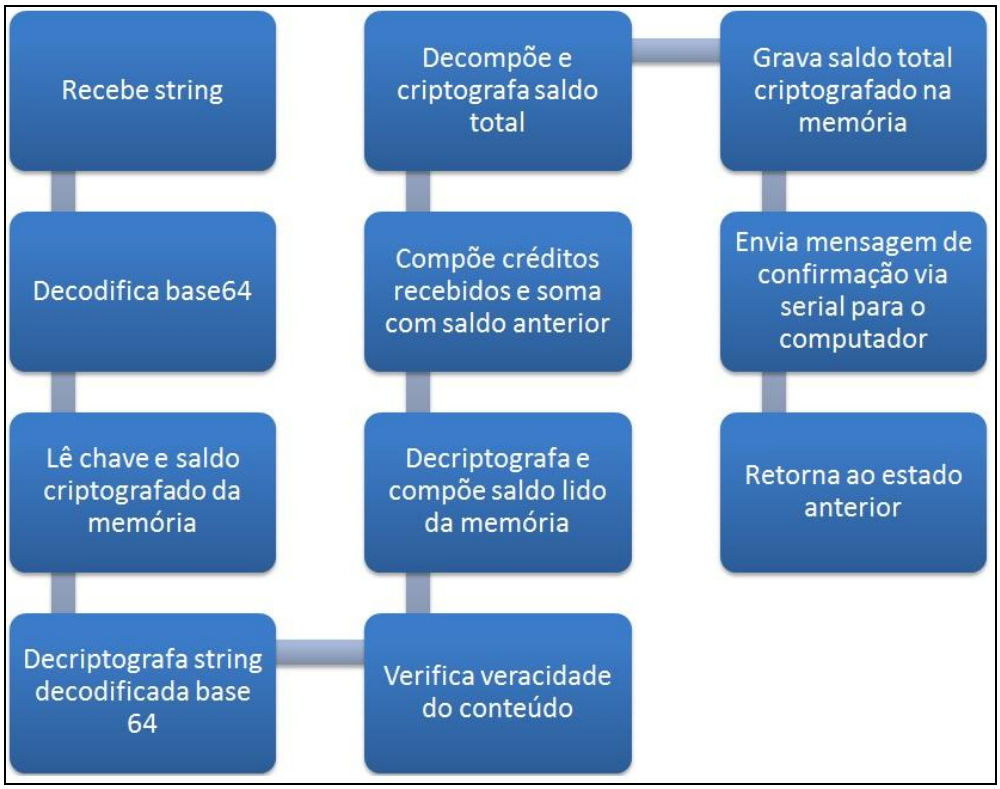


Figura 29 – Processo de recepção e armazenamento de créditos
Fonte: Autoria própria

Caso o menu selecionado seja o “Inicia USB?” e o usuário selecionar o botão de navegação, o horário atual será mostrado. A partir desse menu, se o usuário apertar a tecla “Selecionar”, entrará no estado em que ele pode ajustar o horário do relógio. Nesse estado o usuário deve selecionar os números de 0 a 23 para horas, de 0 a 59 para minutos e 0 a 59 para segundos com o botão de navegação, seguido do botão “selecionar” quando selecionado o número desejado. Assim que finalizado o ajuste de horário, o microcontrolador retornará ao estado de exibição das horas. Depois disso, se o usuário pressionar novamente o botão de navegação, retornará ao estado em que os créditos são mostrados, fechando assim o ciclo.

3.2.1 Protocolo I2C e interface UART

O MSP430 disponibiliza três módulos diferentes para a comunicação serial o USART, USCI e USI [2]. Especificamente o protocolo I2C só é disponibilizado nos módulos USCI e USI. Como o módulo USI exige uma máquina de estados desenvolvida pelo usuário, utilizamos o USCI que é um módulo mais sofisticado e de mais fácil utilização. O módulo USCI ainda é divido em USCI_A e USCI_B. Sendo o primeiro para utilização da interface UART, IrDA e SPI e o segundo para I2C e SPI. Portanto foi utilizado o modo USCI_A para o desenvolvimento da interface UART e o USCI_B para o protocolo de comunicação I2C.

O USCI permite que se trabalhe com os modos de baixo gasto energético do MSP430, trabalhando com o I2C na base interrupções ao microcontrolador. Primeiramente é preenchido um vetor com todos os bytes a serem enviados na comunicação, e ao ser inicializado o módulo a partir de seus registradores, o microcontrolador entra em estado de baixo gasto energético e só volta a ficar ativo quando for requisitado.

A maior dificuldade encontrada pela equipe para o funcionamento do módulo USCI_B para I2C foi o armazenamento do endereço correto no registrador que requer o *Slave Address*. Como a sequência de leitura e escrita no protocolo I2C é diferente, o módulo USCI não precisa do ultimo bit de controle, geralmente acompanhado do *Slave Adress*, que é “1” para quando se deseja ler e “0” quando se deseja escrever no *Slave*. Por isso, o endereço a ser armazenado no registrador é o *Slave Adress* rotacionado um bit para a direita. Por exemplo, o endereço para leitura do PCF8563 é 0xA3 e para leitura 0xA2. No modulo USCI, deve-se guardar o valor 0x51 no registrador UCB0I2CSA.

Foi utilizado o USCI_A para o desenvolvimento da interface UART. Porém, não foi houve dificuldade em sua aplicação devido à enorme variedade de exemplos disponíveis na internet. Foram desenvolvidas funções de recepção e transmissão de caracteres.

3.2.2 Visor OLED

Após o funcionamento correto do protocolo I2C foi possível iniciar a programação do visor OLED. Como o visor seria utilizado com diversas vezes, foi desenvolvida uma biblioteca com as funções que seriam mais utilizadas. Entre elas:

- *Init_oled()* – Inicializará todos os parâmetros necessários para o correto funcionamento do display;
- *escreve_oled(char posicao, char *pstr)* – O primeiro parâmetro dessa função recebe o endereço da posição onde que se deseja imprimir, e o segundo recebe a string a ser impressa no visor OLED;
- *display_time(char posição, unsigned char *time_p);* - Similar a função escreve_oled, mas como os dados lidos do RTC estão em formato BCD, na função *display_time* foi desenvolvido a conversão de BCD para ASCII para que pudesse ser legível no display;
- *Display_creditos(char posição, int16_t cred_var)* - Similar a função *display_time*, mas nela os créditos estão em forma hexadecimal e são do tipo int16 ao invés de “char” para que seu limite máximo não seja apenas 255 e sim 65535 minutos de crédito. Nessa função, então, é convertido o valor em hexadecimal para ASCII para uma visualização legível dos créditos no visor.

3.2.3 Relógio de tempo real

O PCF8563 disponibiliza funções de *timer* e alarme. No *timer*, o programador pode definir quanto tempo a partir da inicializaçãoo PCF8563 irá emitir um sinal de interrupção através do pino 3. Já no alarme, o usuário pode definir que horas, minutos, dia da semana, dia, mês e ano precisamente o PCF8563 irá emitir um sinal de interrupção. Neste projeto foi utilizada a função *timer* com um tempo de 60 segundos, isto é, a partir do momento que o usuário iniciar o desconto de créditos, um timer de 60 segundos é ativado, e durante cada interrupção do RTC o

microcontrolador irá descontar um crédito e inicializar novamente um *timer* de 60 segundos.

Assim como no OLED, foi desenvolvida também uma biblioteca que comportasse todas as funções que utilizassem o RTC. As principais funções dessa biblioteca são:

- *write_time(unsigned char seconds_var, unsigned char minutes_var, unsigned char hours_var, unsigned char day_var, unsigned char weekday_var, unsigned char months_var, unsigned char year_var)* – Armazena a data recebida nos parâmetros no RTC;
- *Read_Time(unsigned char *time_p)* – armazena no vetor enviado como parâmetro a data contida no RTC;
- *Start_Timer(unsigned char tempo_countdwon)* – Define a quantidade de tempo em segundos no qual iniciará o *timer*, e.g.,Start_Timer(60) despertará a interrupção no PCF8563 60 segundos depois da execução dessa função;
- *Clear_Timer(void)* – Retira o *timer* que estiver inicializado no RTC.

3.2.4 Criptografia AES128

Como foi utilizada uma biblioteca fornecida pela própria TI [23], não houve muita dificuldade em seu uso. A biblioteca acompanha um manual explicando como utilizar as funções, sendo que deve-se ter atenção para o fato de que as chaves e as palavras utilizadas na criptografia precisam ter exatamente 128 bits, ou 16 bytes. Dessa forma, foi utilizada uma combinação para distribuir a variável de crédito em um vetor de 16 bytes. Uma função é utilizada para todo o processo:

- aes_enc_dec(unsigned char *state, unsigned char *key, unsigned char dir) – Os dois primeiros parâmetros são a palavra a ser codificada ou decodificada e a chave utilizada, respectivamente. O terceiro parâmetro deve ser 0 para criptografar e 1 para decriptografar. Vale ressaltar que toda vez que se usa uma chave para criptografar, essa chave é modificada pela função. Assim, é necessário fazer uma cópia da chave para que não a perca no processo de criptografia.

3.2.5 Memória *Flash*

A memória *flash* do MSP430 tem como função principal guardar o código desenvolvido. Mas devido à flexibilidade no seu design é possível escrever e ler informações na memória *flash* em tempo de execução. Isso permite escrever e ler informações que persistam mesmo depois de desligar o microcontrolador [27]. No projeto em questão, é essencial a utilização da memória *flash* para se armazenar a chave de criptografia de um dispositivo e também a quantidade de créditos, dispensando a utilização de uma memória EEPROM externa.

Existem quatro segmentos de memória de 64 bytes no MSP430G2553 os quais são livres para o uso geral do programador [2]. Um segmento de memória é a parcela mínima de informação que pode ser gravado na memória, ou seja, mesmo que se deseje armazenar apenas 3 *bytes*, no processo de gravação um segmento inteiro de 64 *bytes* será apagado. No entanto, para poder ler informações da memória *flash*, basta utilizar um ponteiro apontando para o endereço de memória onde se localiza a informação e recuperar os dados. A segmentação da memória *Flash* no MSP430G2553 pode ser vista na Figura 30.

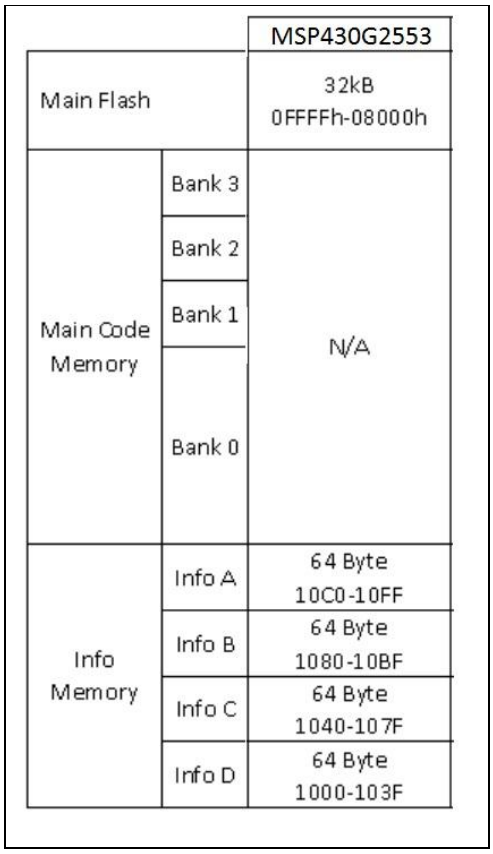


Figura 30 – Segmentação da memória do *flash* do microcontrolador MSP430G2553
Fonte: LITOVSKY, Gustavo [27]

Neste projeto, o conteúdo a ser armazenado na memória foi a chave de criptografia e decriptografia e a quantidade de créditos criptografada. Ambos possuem a mesma extensão: 16 *bytes*. Porém, foi utilizado um segmento de memória para cada uma dessas informações para que ficassem desacopladas, uma vez que o conteúdo de créditos criptografados é constantemente alterado e a chave de usuário é fixa.

3.3 *WEBSITE*

O *website* foi desenvolvido em linguagem PHP e HTML e utiliza como banco de dados o MySQL. É dividido em duas partes, a administrativa e a área do cliente. Na área administrativa, somente os administradores têm acesso, e é onde é possível

adicionar e editar novos produtos, usuários, outros administradores e aprovar compras pendentes.

No banco de dados do site é feito o gerenciamento das chaves de criptografia. Foi criada uma tabela que relaciona o número “serial”, que seria um número único que acompanha cada dispositivo embarcado, com a chave utilizada no dispositivo embarcado. Esse número serial é inserido pelo cliente no momento de seu cadastro no *website*.

Após a compra do pacote de créditos desejado, é enviado um pedido para a área de administração. Qualquer administrador do site pode aceitar o pedido, e a partir desse momento a quantidade de créditos comprada é somada com a quantidade de créditos já existente na conta do cliente. Após a soma, os créditos são criptografados, armazenados e associados ao cliente. Depois de aprovada a compra, a qualquer momento o usuário pode transferir todos os créditos que possui.

3.3.1 Criptografia AES 128 no *website*

Para a criptografia no *website* foi utilizada a biblioteca AES 128 para PHP [28]. No processo de criptografia ainda há a utilização de um codificador em base64. Codificação em base 64 é útil para transformar um vetor de números não imprimíveis em um vetor com caracteres texto. Isto tornou-se necessário pois o banco de dados não oferece suporte para armazenar vetores hexadecimais, sendo que as informações criptografadas podem assumir qualquer número hexadecimal de oito bits. O processo de codificação em base64 consiste em dividir a palavra a ser criptografada de seis em seis bits. A cada seis bits, uma nova contagem é feita a qual pode ser de 0 até 63 (do binário 111111). É feita uma consulta com o número formado em uma tabela pré-existente (Tabela 8), que associa um número a um caractere imprimível [29].

Tabela 8: Alfabeto da codificação base64

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

Fonte: TSCHABITSCHER, Heinz [29]

Ainda no *website* foi desenvolvido um bloco na linguagem *javascript* para que fosse possível realizar a comunicação com a extensão.

3.4 EXTENSÃO E APLICATIVO

A extensão foi desenvolvida nas linguages JavaScript, HTML e CSS, utilizando como base a extensão pública "messaging" disponibilizada no repositório *GitHub* [30]. Ela é composta de um arquivo *manifest* e de um *script* em javascript que roda em plano de fundo. Neste arquivo, sua função principal é receber mensagens externas (vindas do aplicativo ou do *website*), interpretá-las e reenviá-las conforme seu significado. Por exemplo, caso a mensagem seja os créditos, ela repassa a *string* criptografada para o aplicativo, caso a mensagem seja a *string* de verificação que o microcontrolador enviou, ela repassa para o *website*. Cada aplicativo e cada extensão possui um identificador (ID), e através deste ID, a extensão identificará daonde a mensagem estará chegando.

O aplicativo também foi desenvolvido nas linguages JavaScript, HTML e CSS, com base no aplicativo público disponibilizado no repositório GitHub "Diginow Serial Terminal" [31]. Sua interface com o usuário consiste na seleção da porta

Serial correspondente ao dispositivo embarcado, por isso o aplicativo deve ser aberto antes da transmissão dos créditos.

Foram utilizadas principalmente as APIs Serial e Message Passing:

- API Serial: ler as mensagens enviadas pelo dispositivo e enviar os créditos para o dispositivo. Um evento é gerado quando algum dado é recebido através da porta serial, e a função *listener* (ouvinte) direciona para a função que interpretará a mensagem. A mensagem enviada é a dos créditos criptografados, e a mensagem recebida esperada é a string "OK" [24].
- API Message Passing: receber e mandar mensagens para extensão. Possui também uma função listener para interpretação da mensagem. A mensagem enviada é a string "OK" vinda do dispositivo (representando que os créditos foram devidamente descontados), e a mensagem recebida é a string dos créditos criptografados vinda do *website* [33].

A extensão pode ser vista na Figura 31 – Extensão Chrome e o aplicativo pode ser visto na Figura 32.

Figura 31 – Extensão Chrome
Fonte: Autoria própria

Figura 32 – Aplicativo Chrome
Fonte: Autoria própria

4 ANÁLISE DE RESULTADOS

Nesta seção é apresentado o funcionamento geral do sistema e são feitas comparações dos resultados esperados propostos na disciplina de TCC1 com os resultados obtidos.

4.1 FUNCIONAMENTO

Cada dispositivo embarcado ao ser adquirido pelo cliente é acompanhado de um número “serial”. Este número serial é um número único e está associado à chave única daquele dispositivo embarcado e é requisitado durante o cadastro do usuário no *website*. O *website* pode ser acessado em http://estareletronico.com/.

Figura 33 – Tela inicial do *website* EstaR Eletrônico
Fonte: Autoria própria

Depois de feito o cadastro, o usuário necessita realizar seu *login* para efetuar a compra de créditos. Os pacotes de créditos disponíveis podem ser vistos no canto esquerdo do *website*. Ao efetuar uma compra, a situação do produto ficará como “Pendente” até que um administrador aprove a transação. Quando algum administrador realizar a aprovação, o usuário poderá ver o saldo atual no *website* ao clicar em “Minhas Compras”. Nessa página é possível ver todas as transações já realizadas, a situação de cada uma delas além do saldo atual.

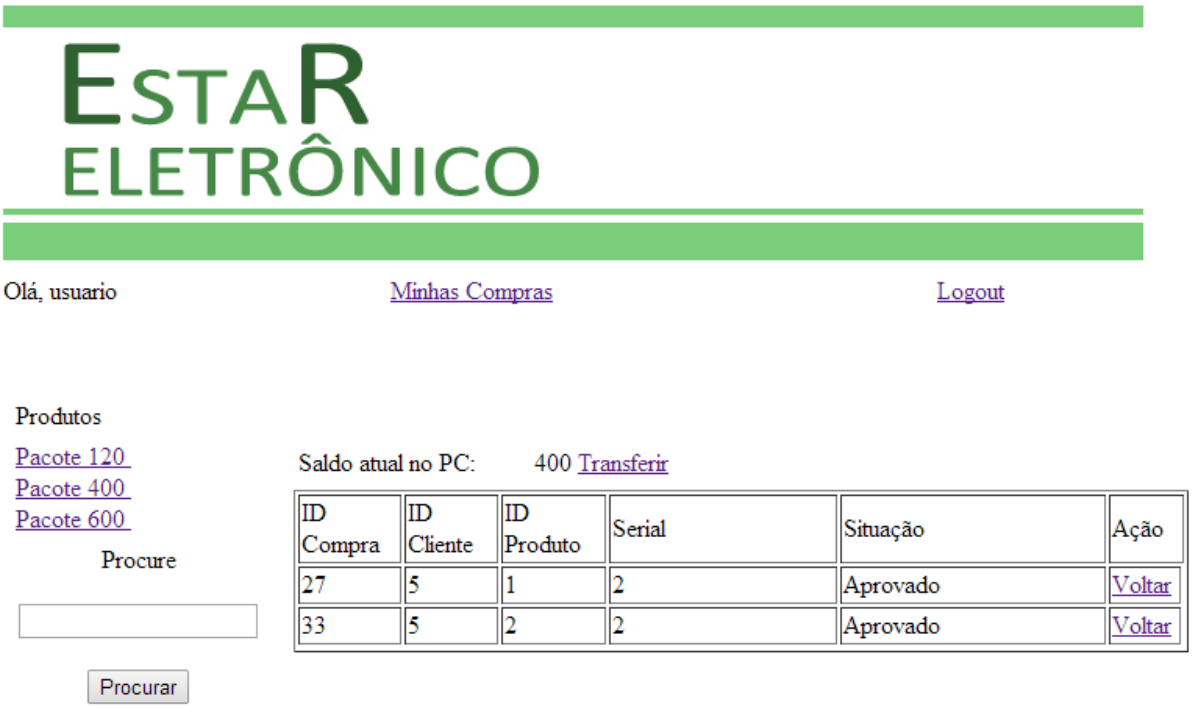


Figura 34 – Página de "Minhas Compras" mostrando saldo atual e compras realizadas
Fonte: Autoria própria

Ao clicar em “transferir” o usuário precisará conectar o dispositivo embarcado na entrada USB de seu computador, selecionar o modo de transferência, abrir a extensão e selecionar a porta serial no aplicativo. Feito isso, basta clicar no botão “Transferir Agora”. Caso a transferência tenha ocorrido com sucesso, ou seja, os créditos tenham sido reconhecidos pelo microcontrolador, o saldo no *website* será zerado e o dispositivo embarcado irá sair do modo de transferência.

Com o dispositivo embarcado contendo créditos, basta o usuário ao estacionar o carro em vagas assinaladas com a placa de EstaR, selecionar o menu de início de desconto, selecionar o número máximo de horas permitidas na vaga, e o desconto de créditos será realizado até que seja pressionado o botão “voltar”, atinja o número máximo de horas permitido ou o saldo se esgote.

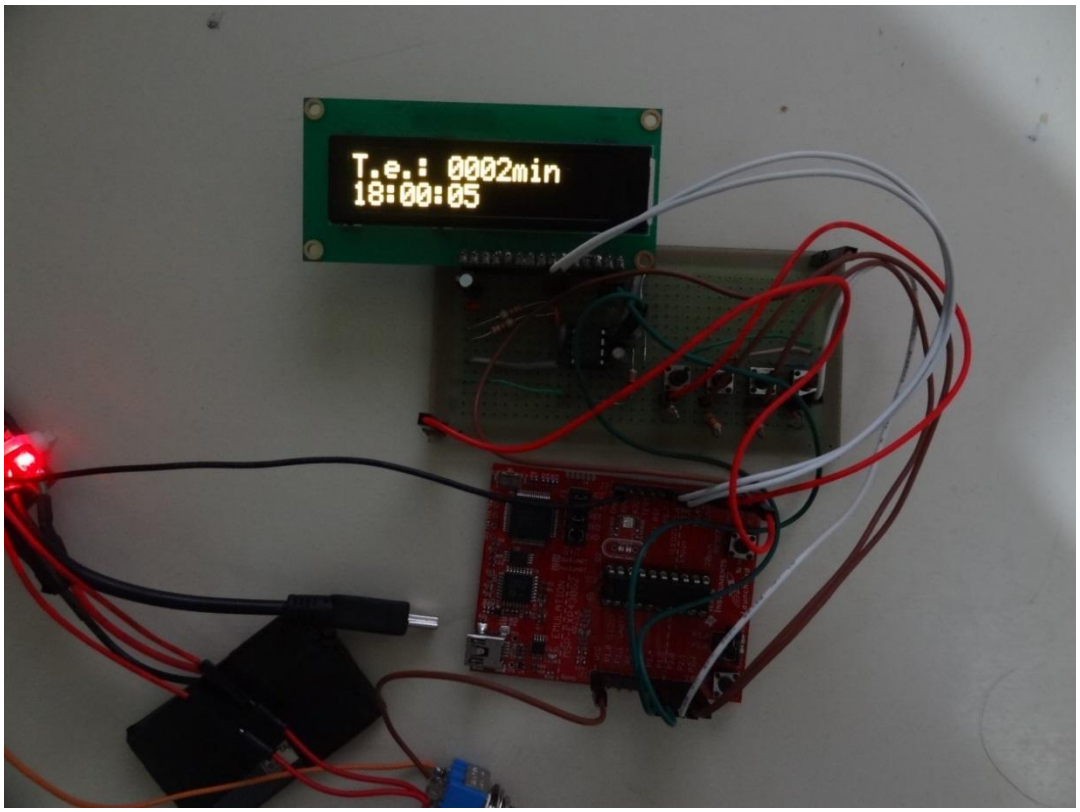


Figura 35 – Dispositivo eletrônico realizando o desconto de créditos (T.E: Tempo estacionado igual a 2 minutos) e início do funcionamento às 18 horas e 5 segundos
Fonte: Autoria própria

4.2 TESTES

Foram realizados testes para mensurar a autonomia do dispositivo. Após a bateria ter sido carregada completamente, mediu-se a corrente utilizada pelo dispositivo e a tensão correspondente com o microcontrolador no estado de desconto de créditos. Depois de três amostras, observou-se um gasto médio de 160mA nesse estado, sendo 130mA somente do display OLED e o restante do microcontrolador MSP430 e do RTC. A curva da Figura 36 de nível de tensão x tempo em minutos foi obtida.

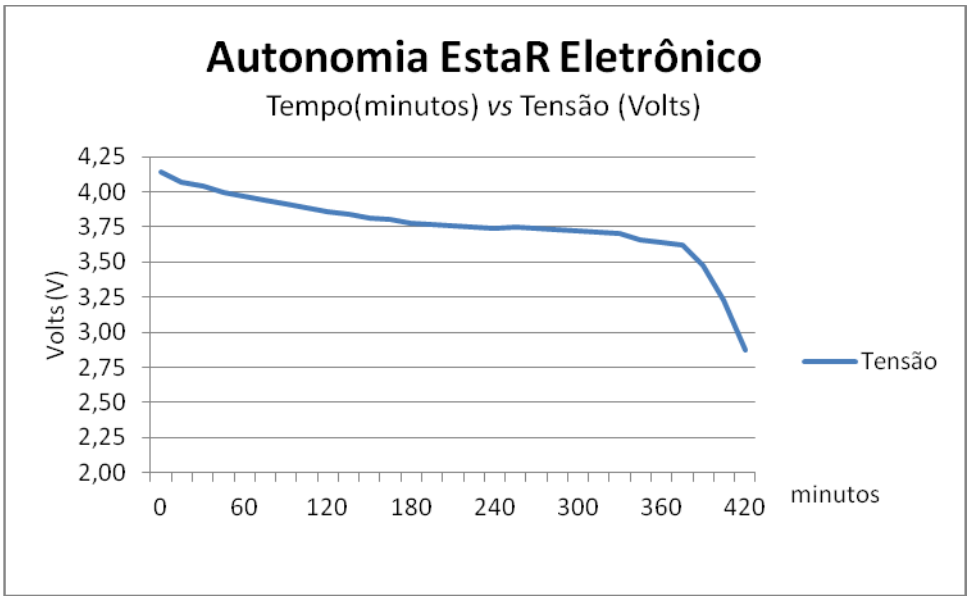


Figura 36 – Curva da autonomia do EstaR Eletrônico
Fonte: Autoria própria

Nota-se que o dispositivo embarcado possui uma autonomia de aproximadamente sete horas de uso contínuo, considerando a bateria de 1200mAh. Essa autonomia é suficiente para a utilização do dispositivo, que pode ser de até no máximo 3 horas de uso (máximo de horas contínuas permitidas em Curitiba). Além disso, a entrada *mini*-USB do módulo de recarga permite que a bateria seja recarregada na saída do acendedor de cigarro do automóvel.

Entretanto, é possível dizer que o visor OLED em questão não possui vantagens energéticas quando comparado com um visor LCD de características semelhantes. Um visor LCD de 16 caracteres e 2 linhas, com *backlight* aceso consome em média 120mA [34]. Ainda, se fosse utilizado o visor LCD sem a luz de fundo acesa, o consumo médio seria de 1.2mA, que somados ao gasto do microcontrolador e do RTC de 30mA, forneceria uma autonomia de até 38 horas de uso contínuo.

4.3 RESULTADOS TECNOLÓGICOS

✓ **Composição de *hardware* portátil, compacto, de fácil manuseio.** Durante todo o projeto buscou-se componentes eletrônicos que fossem que

proporcionassem o desenvolvimento de um dispositivo portátil. Se dispostos em uma caixa recipiente, fica prático para o usuário final transportar seu dispositivo embarcado quando necessário. Quanto à facilidade do manuseio, é necessário realizar uma campanha de testes e validações com os usuários para determinar a melhor maneira de apresentação de dados, seleção de menus e início de funcionamento. Mesmo assim, o dispositivo embarcado é relativamente simples e são necessários somente três botões de ajustes para todas as funcionalidades, sendo para que para acionar a função principal, é necessário apenas pressionar um botão.

- ✓ **Composição de *software* específico para recarga do dispositivo**. Foi construído um *website* para a realização das compras de créditos, de cadastro de clientes e associação das chaves de criptografia. Além disso, foi ainda desenvolvida uma maneira de enviar os créditos criptografados do *website* para o dispositivo embarcado com a construção de um aplicativo e uma extensão na plataforma Chrome.
- ✓ **A aplicação de algoritmos de criptografia no número de créditos dos usuários, para que o sistema se torne mais seguro e mais difícil de ser fraudado.** O algoritmo de criptografia AES 128 foi utilizado para a criptografia do saldo no *website* e para decriptografia de créditos no microcontrolador. O gerenciamento de chaves foi realizado de forma eficaz, possibilitando que cada usuário tenha uma chave específica, tornando o sistema mais difícil de ser fraudado.

4.4 RESULTADOS CIENTÍFICOS

- ✓ **Desenvolvimento de um projeto que envolva os conhecimentos adquiridos durante o curso.** Foi adquirido um conhecimento mais profundo do microcontrolador MSP430, do protocolo de comunicação I2C, da interface UART e da utilização da memória *flash* para armazenar valores persistentes. Ainda envolveu conhecimentos além dos adquiridos durante o curso, para a construção do aplicativo, da extensão através da linguagem JavaScript, do *website* através das linguagens de programação HTML, PHP e do banco de dados utilizando MySQL.

4.5 RESULTADOS ECONÔMICOS

- ✓ **Desenvolvimento de um sistema que minimize o gasto com a impressão dos blocos de cartões de papel a médio e longo prazo e melhore a relação de custo e benefício para o usuário com um sistema de cobrança diferenciado, substituindo o sistema "por hora" com créditos descontados por minuto.** Com a inicialização da contagem dependendo apenas de um único botão, diminui-se a quantidade blocos de papel desperdiçados devido a erros de marcação do dia e horário a ser estacionado. O sistema desenvolvido faz o desconto de minuto a minuto, beneficiando o usuário que não utiliza a hora cheia. Como um bloco de papel é jogado fora toda vez que é usado, estima-se que o dispositivo embarcado, mesmo tendo um custo mais elevado em curto prazo quando comparada com um talão de blocos, em longo prazo custe menos devido à virtualização dos créditos.

4.6 RESULTADOS SOCIAIS

- ✓ **Desenvolvimento de um sistema que garanta a rotatividade e a democratização do espaço público**. O sistema desenvolvido de estacionamento regulamentado eletrônico mantém a eficácia em relação à rotatividade e democratização do espaço público do sistema de blocos.

4.7 RESULTADOS AMBIENTAIS

- **Diminuir o impacto ambiental causado pela emissão de blocos de cartões de papel.** Com a utilização de bateria recarregável e componentes da diretriz RoHS (Restriction of Hazardous Substances) que prevê a eliminação da utilização substâncias tóxicas ao meio ambiente na produção de equipamentos eletrônicos, é possível utilizar um sistema mais sustentável, entretanto dificilmente irá diminuir o impacto ambiental de emissão de blocos de cartões de papel que podem ser reciclados.

5 GESTÃO DO PROJETO

5.1 CRONOGRAMA

O cronograma de desenvolvimento do projeto sofreu um atraso de quatro meses no início da etapa de desenvolvimento. O cronograma planejado pode ser visto na Tabela 9 e o cronograma executado na Tabela 10.

Tabela 9 : Cronograma planejado

Tarefas	Período, em meses											
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar
Definição do tema do projeto													
Questionamentos e pesquisas sobre o tema													
Definição dos requisitos													
Pesquisas dos componentes													
Estudos do desenvolvimento													
Efetuar a compra dos componentes													
Implementação dos periféricos													
Programação do dispositivo													
Desenvolvimento do site e/ou *software*													
Interface entre dispositivo e computador													
Testes													
Finalização da documentação													
Apresentação													

Fonte: Autoria própria

Tabela 10: Cronograma realizado

Tarefas	Período, em meses												
	2013										2014		
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar
Definição do tema do projeto													
Quesetionamentos e pesquisas sobre o tema													
Definição dos requisitos													
Pesquisas dos componentes													
Estudos do desenvolvimento													
Efetuar a compra dos componentes													
Integração dos periféricos													
Programação do dispositivo embarcado													
Composição do protótipo do hardware													
Desenvolvimento do site e/ou software													
Interface entre dispositivo e computador													
Composição final do hardware													
Testes													
Finalização da documentação													
Apresentação													

Fonte: Autoria própria

O atraso no início da etapa de desenvolvimento foi devido à carga horária elevada do semestre letivo naquele período para ambos os alunos envolvidos no projeto. Além disso, o tempo de desenvolvimento da interface entre o dispositivo e o computador tomou mais tempo do que o programado, uma vez que, a comunicação entre *websites* e dispositivos embarcados pela porta serial ou USB não é comum por motivos de segurança.

A soma desses imprevistos teve impacto significativo no desenvolvimento do projeto como um todo, resultando em um acúmulo de tarefas em sua etapa final.

5.2 ANÁLISE DE CUSTO DO PROJETO

Foi utilizada a taxa de câmbio de um dólar igual a R$2,20 e também foi considerado o preço do produto na época da compra.

5.2.1 Custos diretos

Tabela 11: Custos diretos da produção de um dispositivo embarcado.

Item	Valor unitário	Unidades por dispositivo	Valor total
Kit LaunchPad EXP460G2	R$ 21,97	1	R$ 21,97
Real Time Clock PCF8563	R$ 1,76	1	R$ 1,76
Visor OLED com CI US2066	R$ 22,99	1	R$ 22,99
Módulo de recarga TP4056	R$ 2,09	1	R$ 2,09
Bateria BL-5C	R$ 9,90	1	R$ 9,90
Placa de cobre	R$ 3,00	1	R$ 3,00
Caixa recepiente	R$ 4,89	1	R$ 4,89
Demais componentes(resistores, capacitores e push buttons)	R$ 2,00	1	R$ 2,00
Total			R$ 68,60

Fonte: Autoria própria

Esses são valores para um único produto isolado. Porém, o custo por unidade do dispositivo embarcado pode ser reduzido pela produção em grandes quantidades. Para produção em maior escala, será considerado 1000 unidades, utilizando, portanto, o preço de atacado:

Tabela 12: Projeção de custos diretos da produção em maior escala

Item	Valor unitário	Unidades	Valor total
Kit LaunchPad EXP460G2	R$ 13,64	1000	R$ 13.640,00
Real Time Clock PCF8563	R$ 0,26	1000	R$ 260,00
Visor OLED com CI US2066	R$ 19,44	1000	R$ 19.440,00
Módulo de recarga TP4056	R$ 1,67	1000	R$ 1.670,00
Bateria BL-5C	R$ 7,67	1000	R$ 7.670,00
Placa de cobre	R$ 2,12	1000	R$ 2.120,00
Caixa recepiente	R$ 3,87	1000	R$ 3.870,00
Demais componentes(resistores, capacitores e push buttons)	R$ 1,12	1000	R$ 1.120,00
Total		1000	R$ 49.790,00
Total por unidade		1	**R$ 49,79**

Fonte: Autoria própria

Nota-se que o custo por unidade do dispositivo embarcado diminui de R$68,60 para R$49,79, que significa uma redução de 27,42% no valor total do produto.

5.2.2 Custos indiretos

Tabela 13: Custos indiretos de produção

Item	Valor unitário	Quantidade	Valor total
Hospedagem de website	R$12,50 por mês	-	R$ 12,50
Analisador de sinais lógicos portátil	R$ 19,48	1	R$ 19,48
Multímetro	R$ 37,42	1	R$ 37,42
Protoboard	R$ 15,90	1	R$ 15,90
Cabos Banana-Jacaré	R$ 6,50	4	R$ 26,00
Ferro de solda	R$ 14,75	1	R$ 14,75
Energia Elétrica	R$ 35,00	-	R$ 35,00
Internet	59,9 por mês	-	R$ 59,90
Computador para programação	R$ 900,00	1	R$ 900,00
Total			R$ 1.120,95

Fonte:Autoria própria

Nota-se um gasto de R$1.120,95 com custos indiretos. Entretanto, a maior parte desses produtos são gastos únicos para que fosse possível realizar o desenvolvimento do produto e não crescem na mesma proporção na produção de cada dispositivo embarcado.

5.3 ANÁLISE DE RISCOS DO PROJETO

O quadro abaixo faz uma comparação dos riscos previstos pela proposta e os problemas que ocorreram durante o desenvolvimento do projeto. Embora nenhum risco diferente tenha ocorrido, os que ocorreram tomaram mais tempo do que o planejado, resultando no atraso e acumulação das tarefas que pode ser vista no cronograma da Tabela 10.

Tabela 14: Análise de riscos do projeto

Grau	Risco	Ação Prevista	Ocorrência	Ação tomada
0,1	Problemas na composição da PCI	Detectar erros e corrigir projeto	Não	-
0,1	Problemas na montagem mecânica do aparelho	Detectar erros e corrigir projeto	Não	-
0,3	Falha de componentes	Substituição dos componentes e/ou pesquisa e compra de novos	Sim	Compra de novo componente
0,3	Problemas na composição firmware	Detectar erros e corrigir firmware	Sim	Pesquisa em comunidades online para discussões de problemas em comum
0,3	Problemas na composição software (PC)	Detectar erros e corrigir software (PC)	Sim	Pesquisa de meios não convencionais para comunicação entre website e microcontrolador
0,5	Problemas no funcionamento devido à temperatura	Novo estudo de impacto da temperatura no funcionamento do sistema, e novo projeto	Não	-

Fonte: Autoria própria

O primeiro risco ocorrido, classificado como falha de componentes, foi a quebra do visor OLED. Prevendo o mal funcionamento ou a quebra dos componentes e considerando a demora da entrega de componentes comprados da China, foram comprados componentens em quantidade maior do que a necessária para o desenvolvimento do projeto. Porém, como o visor foi danificado logo na etapa inicial do cronograma, foi possível efetuar uma nova compra, para que, ao longo do projeto, ainda fosse mantida uma margem de segurança caso algum outro visor fosse quebrado novamente.

O risco previsto de problema na composição do *firmware* ocorreu diversas vezes. Sempre se buscou analisar a melhor alternativa de implantação do código. Um deles, por exemplo, foi citado neste relatório no capítulo 3.2.1, que foi a utilização correta do módulo USCI do MSP430. Sempre que algum problema ocorria, pesquisas eram feitas na internet, principalmente em comunidades online específicas de microcontroladores. Na maioria das vezes foi possível encontrar outros programadores que passaram pelos mesmos problemas e uma discussão era feita por diversos outros programadores para encontrar uma solução. Assim, investigaram-se as soluções que mais se adaptavam ao projeto.

Problemas na composição do *Software* para o PC aconteceram em duas partes. A primeira era a construção de um *website* em linguagem PHP, HTML e JavaScript e MySQL, sendo que nenhum dos integrantes da equipe tinha conhecimento prévio de alguma dessas linguagens de programação. Por isso, foi

dedicado mais tempo de estudo para que fosse possível a construção do *website*. A segunda parte era da comunicação entre *website* e o microcontrolador. Depois de muita pesquisa, descobriu-se que não há uma maneira direta ou convencional de realizar a troca de informações entre *website* e um *hardware* por questões de segurança. A busca por maneiras alternativas ou parceladas de desenvolvimento dessa interface e seu próprio desenvolvimento tomou mais tempo do que o previsto.

Embora no quadro acima a ocorrência de problemas no funcionamento devido à temperatura não esteja assinalada como positiva, não se tem certeza desse risco. Isso se deve ao fato de que altas temperaturas geralmente tem impacto na vida útil dos componentes eletrônicos. Não foi possível testar e comparar, por exemplo, um equipamento operando em temperatura ambiente e outro sob alta temperatura até que se danifiquem e fazer o levantamento da diferença de vida útil entre eles, pois não houve tempo suficiente desde o término de montagem do dispositivo até a data de confecção deste relatório. Entretanto, foram pesquisados e utilizados componentes que suportassem até pelo menos 70ºC no seu intervalo de temperatura de operação.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal construir uma alternativa tecnológica ao sistema de blocos de cartões utilizados para democratização do espaço público utilizado como estacionamento para carros. Nessa alternativa tecnológica, foi buscado proporcionar praticidade e comodidade aos motoristas usuários de vagas de EstaR. Buscou-se desenvolver um aparelho eletrônico, no qual pudesse ser abastecido com créditos através da compra de créditos pela internet e transferência destes utilizando um computador. Neste sistema desenvolvido, cada crédito equivale a um minuto, e, ao estacionar, o motorista deve iniciar o desconto desses créditos. Assim, o usuário não precisa prever quanto tempo irá permanecer na vaga, e pagará somente pelo tempo real estacionado. Ainda, para se tornar um sistema mais seguro contra fraudes, foi proposta a utilização de algoritmos de criptografia para proteger a quantidade de créditos dos usuários.

O desenvolvimento do sistema foi possível de ser realizado e a maioria das especificações propostas foi cumprida. Os resultados principais obtidos foram:

- aparelho eletrônico com autonomia energética de aproximadamente sete horas, mais do que suficiente para o tempo de utilização de no máximo três horas de estacionamento regulamentado em Curitiba;
- desconto de créditos minuto a minuto, com função principal iniciada apenas por um botão;
- bateria pode ser recarregada no próprio automóvel;
- custo de produção que pode chegar até R$49,79;
- *website* no qual o cliente faz seu cadastro e realiza a compra de créditos;
- desenvolvimento de um sistema em que cada dispositivo eletrônico é associado a um único usuário a fim de que cada usuário possua uma chave única de criptografia;
- meio de transferência de dados entre *website* e microcontrolador através do desenvolvimento de um aplicativo e uma extensão Chrome.

6.1 PERSPECTIVA PARA TRABALHOS FUTUROS

Mesmo com a maior parte dos objetivos atingidos, trata-se de um projeto com possíveis melhoramentos em trabalhos futuros. Como perspectivas, podem ser listadas:

- aperfeiçoameto do código do microcontrolador e escolha de componentes que melhorem significativamente a autonomia do dispositivo embarcado;
- aperfeiçoameto da segurança do sistema, aplicando algoritmos de criptografia mais complexos ou utilizando um microcontrolador com módulo *anti-tamper* como o MSP430F6779;
- atualização do relógio do dispositivo eletrônico com o relógio do computador no momento da conexão do cabo *USB*;
- desenvolvimento de uma maneira de transferir os créditos do computador para o aparelho eletrônico sem que o usuário precise selecionar o modo de transferência, a fim de tornar o dispositivo mais fácil de manusear.
- desenvolvimento de um meio mais simples para o usuário para transferência de dados entre o *website* e o aparelho eletrônico.

REFERÊNCIAS

[1] WORDLE, **Wordle TM**. Disponível em: < http://www.wordle.net/>. Acesso em 13 mar. 2014.

[2] TEXAS INSTRUMENTS, **MSP430G2x53 MSP430G2x13 Mixed Signal Microcontroller**,revisão J, SLAS735J, 2011.

[3] TEXAS INSTRUMENTS, **MSP-EXP430G2 LaunchPad Evaluation Kit,** revisão E, SLAU318E, 2010

[4] IAR SYSTEMS, **IAR Embedded Workbench for MSP430**,2013. Disponível em: < http://supp.iar.com/Download/SW/?item=EW430-EVAL>. Acesso em 12 jul. 2013.

[5] TEXAS INSTRUMENTS WIKI, **Code Composer Studio**, versão 5, 2013. Disponível em: < http://processors.wiki.ti.com/index.php/Download_CCS> Acesso em 12 jul. 2013.

[6] NXP SEMICONDUCTORS, **Real-time clock/calendar PCF8563**, revisão 10, 2012.

[7] SENKOVSKI, Antônio, KOMARCHESQUI, Bruna, **Curitiba bate novo recorde e tem dia mais quente da história**, Disponível em < http://www.gazetadopovo.com.br/vidaecidadania/conteudo.phtml?id=1445622> Acesso em 21 fev. 2014.

[8] ROSAS, Bruno, **Temperatura de carro parado no sol pode chegar a 70 graus**, 2014. Disponível em <http://classificados.folha.uol.com.br/veiculos/2014/02/1413020-temperatura-de-carro-parado-no-sol-pode-chegar-a-70-graus-diz-estudo.shtml>. Acesso em 21 fev 2014

[9] OLED ASSOCIATION, **Continued Advances in OLED Technologies** , 2010.

[10] OLED ASSOCIATION, **OLED Myths & Misunderstandings**, 2009. Disponível em < http://www.oled-a.org/images/pdfs/OLED%20Myths.pdf> Acesso em 10 mar. 2014.

[11] LCD-MODULE, **The New OLED Displays,** 2011. Disponível em <http://www.lcd-module.com/products/oled.html>. Acesso em 21 ago 2013

[12] NEWHEAVEN DISPLAY, **Temperature Compensation for LCDs**, Disponível em < http://www.newhavendisplay.com/tempcomp.html> Acesso em 21 ago. 2013

[13] EASTRISING , **Character OLED Display Module 1602** , Disponível em <http://www.buydisplay.com/default/16x2-character-oled-display-module.html>Acesso em 23 jul 2013

[14] EAGER, Jon, **Battery Options For Embedded Applications**, 2010, Disponível em: < http://electronicdesign.com/power/battery-options-embedded-applications> Acesso em 16 dez.2013

[15] NANINJING TOP POWER ASIC, **TP4056 1ª Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8**, Disponível em < http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf> Acesso em 12 fev. 2014

[16] ROBOT ELECTRONICS, **Using the I2C Bus**, Disponível em < http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html> Acesso em 16 fev.2014

[17] TEXAS INSTRUMENTS, **USB to Serial Port Controller TUSB3410, TUSB310I Data Manual**, 2010, revisão H, SLLS519H. Disponível em < http://www.ti.com/lit/ds/symlink/tusb3410.pdf> Acesso em 12 fev. 2014

[18] FRANK, Durda, **Serial and UART tutorial**, 2013 revisão 43184. Disponível em < http://www.freebsd.org/doc/en/articles/serial-uart/> Acesso em 15 jan. 2014.

[19] MOECKE, Cristian **Criptography for Newcomers.**Disponível em < http://www.cristiantm.com.br/artigos/criptografia> Acesso em 02 fev 2014

[20] NETWORK SORCERY, **Encryption algorithms**. Disponível em < http://www.networksorcery.com/enp/data/encryption.htm> Acesso em 02 fev 2014.

[21] GALVAO, Junior, **Diferença entre chaves simétrica e assimétrica**, 2007. Disponível em <http://pedrogalvaojunior.wordpress.com/2007/11/16/diferencas-entre-chaves-simetrica-e-assimetrica-para-criptografia/> Acesso em 02 fev. 2014,

[22] PORNIN Thomas, **Comparsion of DES, 3DES and AES**, 2011. Disponível em < http://stackoverflow.com/questions/5554526/comparison-of-des-triple-des-aes-blowfish-encryption-for-data> Acesso em 08 jan 2014.

[23] TEXAS INSTRUMENTS, **AES128 – A C Implementation of Encryption and Decryption**, 2009, revisão A, SLAA397A.

[24] PANTECH SOLUTIONS, **Chrome Serial API -Launched**, Disponível em < https://www.pantechsolutions.net/blog/chrome-serial-api-launched> Acesso em 03 mar. 2014.

[25] CHROME, **What are Chrome Apps?**, Disponível em < https://developer.chrome.com/apps/about_apps> Acesso em 03 mar. 2014.

[26] TEXAS INSTRUMENTS, **Finite State Machines for MSP430,** 2008, revisão A, SLAA402A.
Disponível em < http://www.ti.com/lit/an/slaa402a/slaa402a.pdf> Acesso em 11 out. 2013

[27] LITOVSKY, Gustavo, **Beginning Microcontrollers With the MSP430**, versão 0.4. Disponível em < http://glitovsky.com/blog/?page_id=21> Acesso em 12 dez. 2013

[28] RODRIGUES, Fernando, **Criptografia AES em objeto PHP**, 2010. Disponível em < http://www.nacaolivre.com.br/php/criptografia-aes-em-objeto-php/> Acesso em 24 jan. 2014.

[29] TSCHABITSCHER, Heinz, **How Base64 Enconding Works**. Disponível em < http://email.about.com/cs/standards/a/base64_encoding.htm> Acesso em 25 jan. 2014

[30] GITHUB, **Messaging**, Disponível em <https://github.com/GoogleChrome/chrome-app-samples/tree/master/messaging> Acesso em 03 mar. 2014.

[31] GITHUB, **Chrome Serial Terminal App**, Disponível em < https://github.com/RIAEvangelist/Chrome-Serial-App> Acesso em 03 mar. 2014.

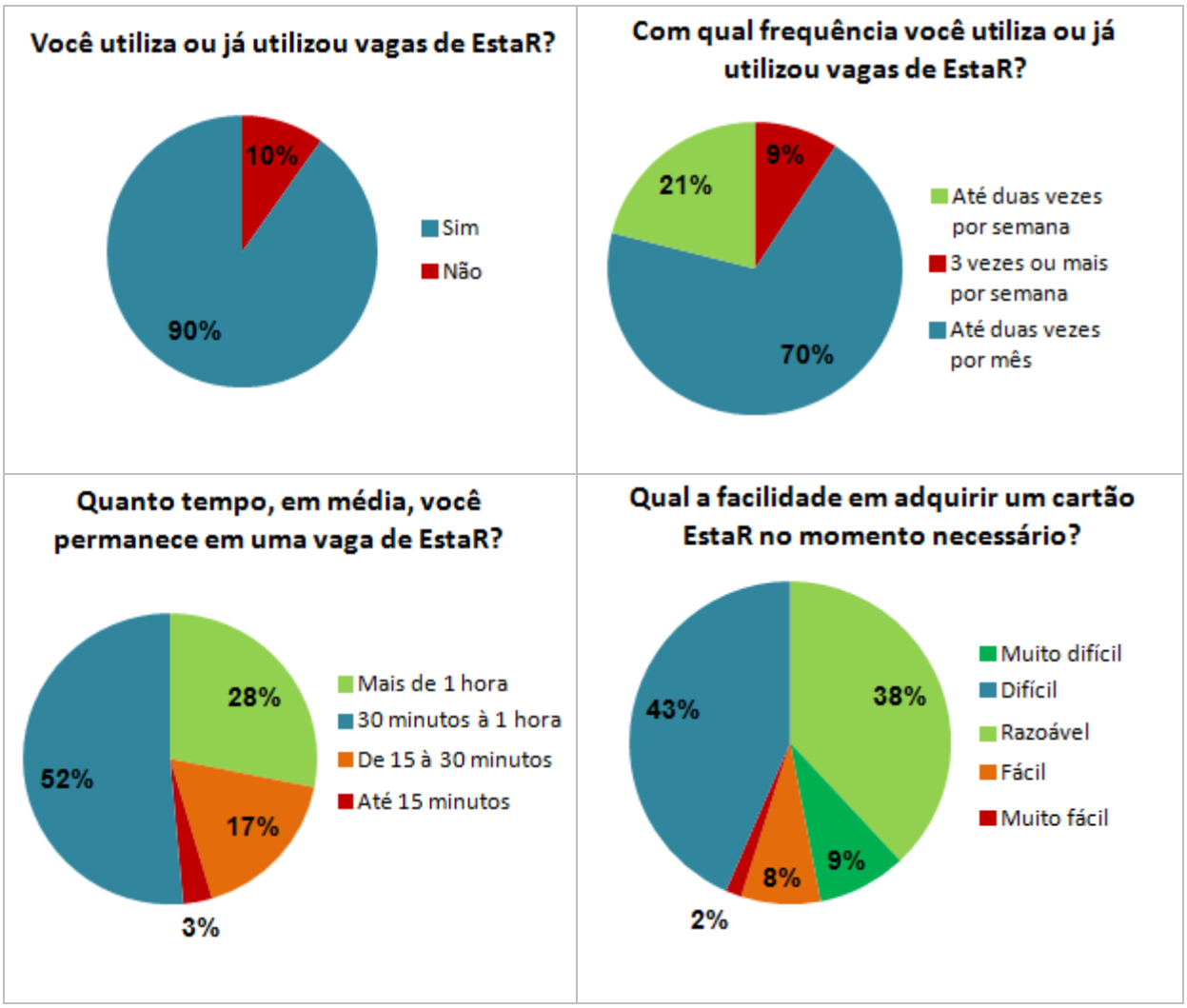
[32] CHROME, **chrome.serial**, Disponível em < https://developer.chrome.com/apps/serial> Acesso em 03 mar. 2014.

[33] CHROME, **Message Passing**, Disponível em < http://developer.chrome.com/extensions/messaging> Acesso em 03 mar. 2014.

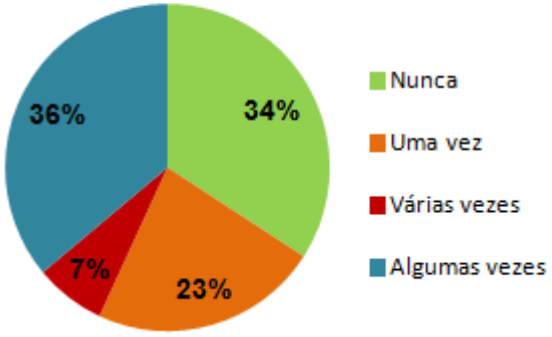
[34] WINSTAR DISPLAY Co, **WH1602B**, Disponível em <http://www.winstar.com.tw/download.php?ProID=22> Acesso em 03 mar. 2014.

APÊNDICES

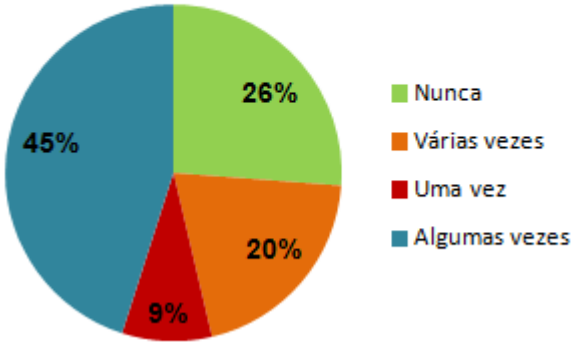
APÊNDICE A — Gráficos da pesquisa de campo



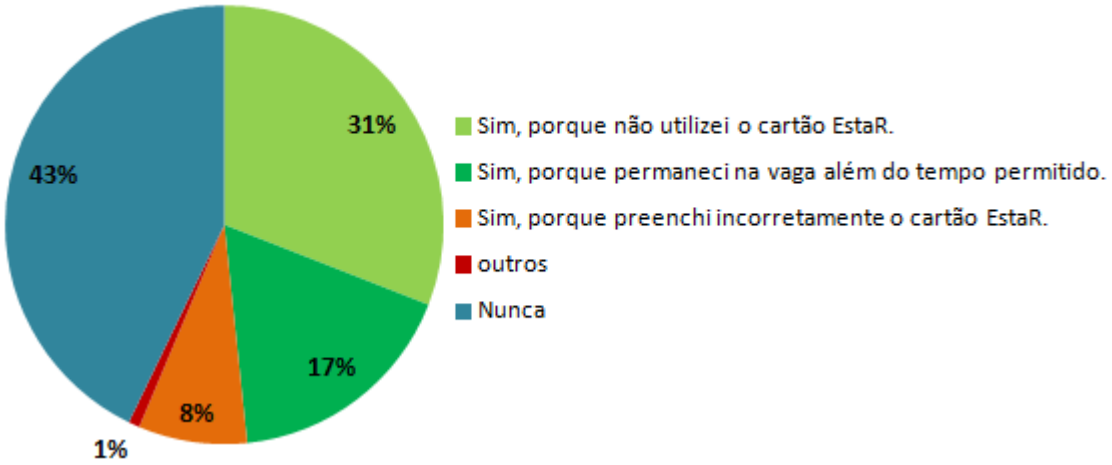
Você já desperdiçou algum cartão EstaR devido a um preenchimento incorreto?



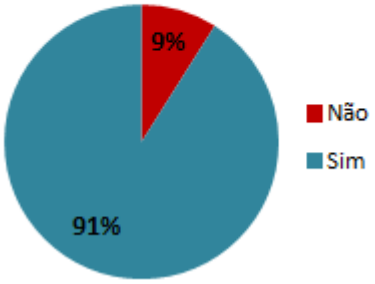
Você já estacionou em uma vaga de EstaR sem utilizar o cartão?



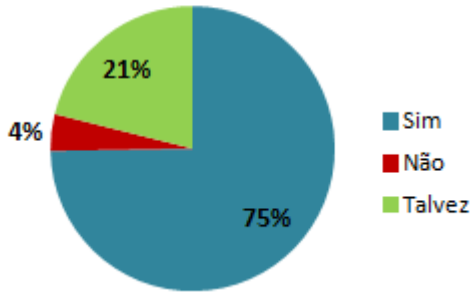
Você já foi notificado ou multado por infração em vaga de EstaR?



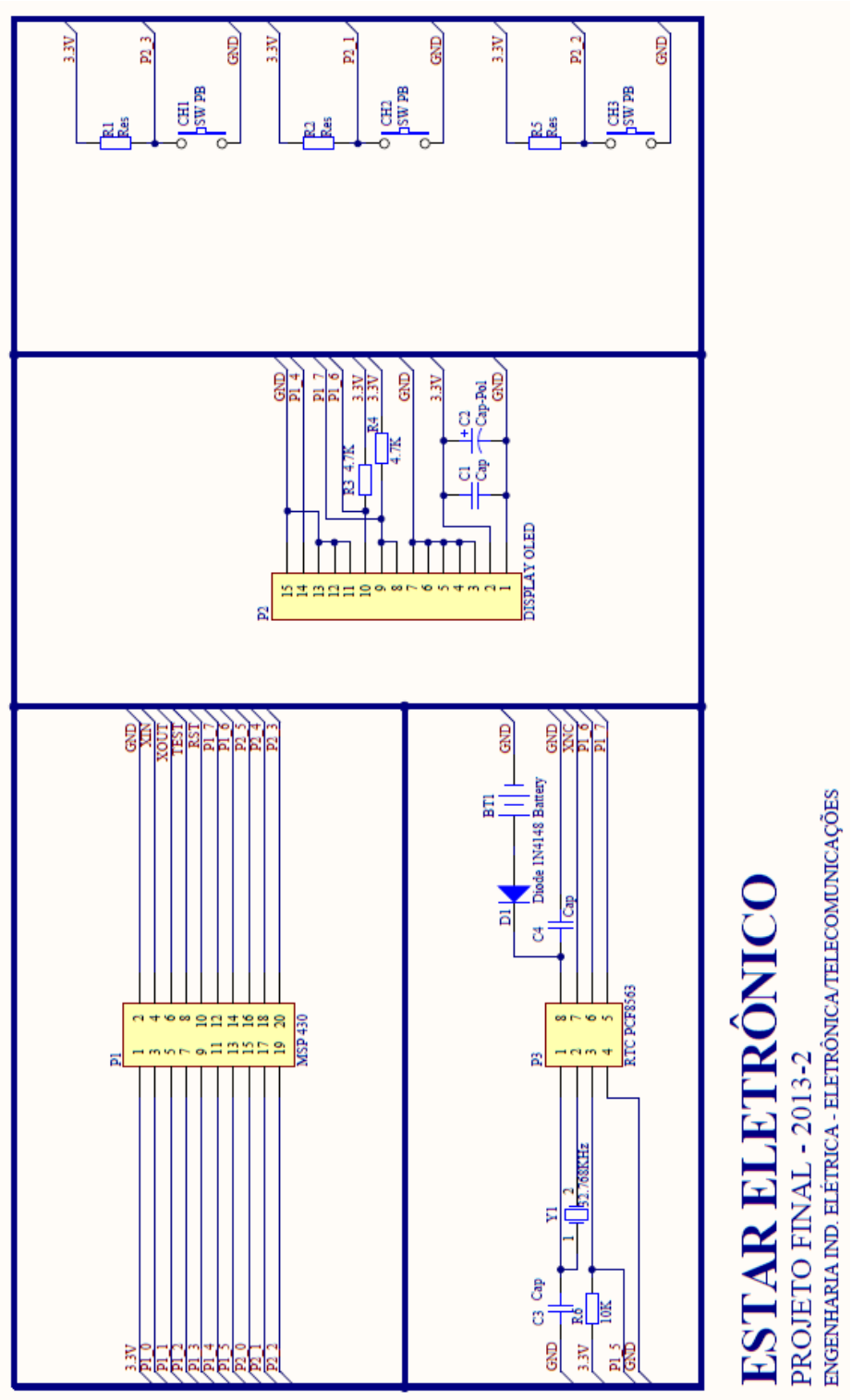
Você preferiria pagar pelo tempo real estacionado em minutos? (ao invés de 1 cartão EstaR para cada 1 hora)



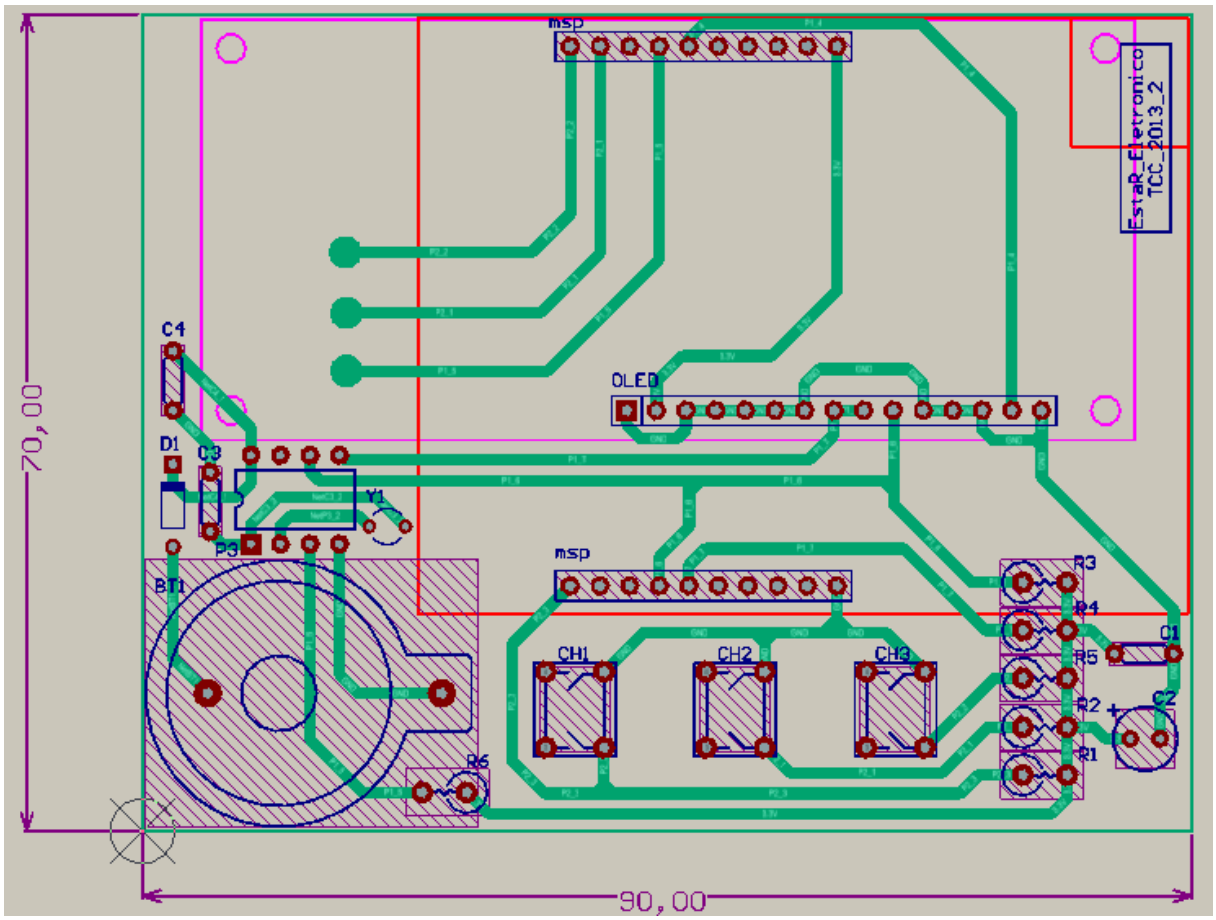
Imagine um aparelho eletrônico, substituto ao cartão EstaR, no qual você poderia recarregar os créditos em casa, sendo que cada crédito equivaleria a 1 minuto de permanência em vagas de EstaR. Você optaria pela utilização deste aparelho ao invés do cartão EstaR?



APÊNDICE B — Esquemático da placa



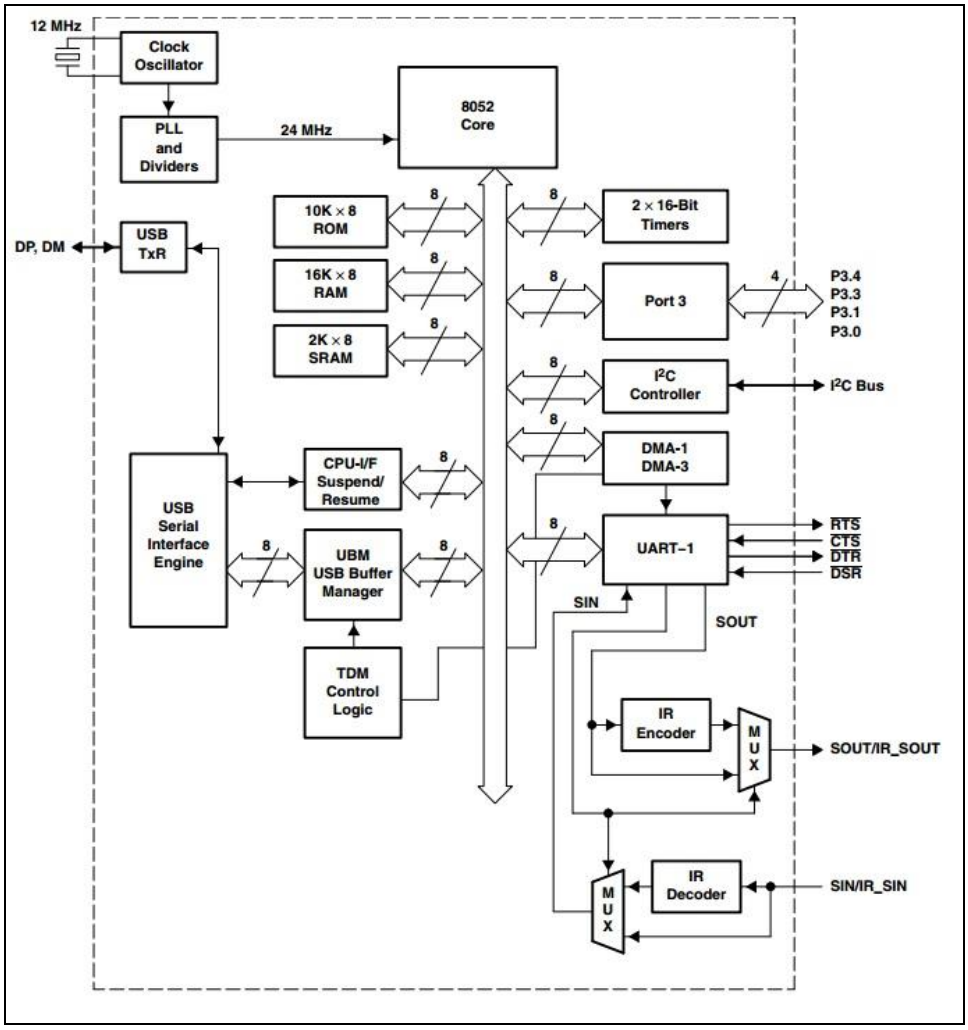
APÊNDICE C — Layout da placa



ANEXOS

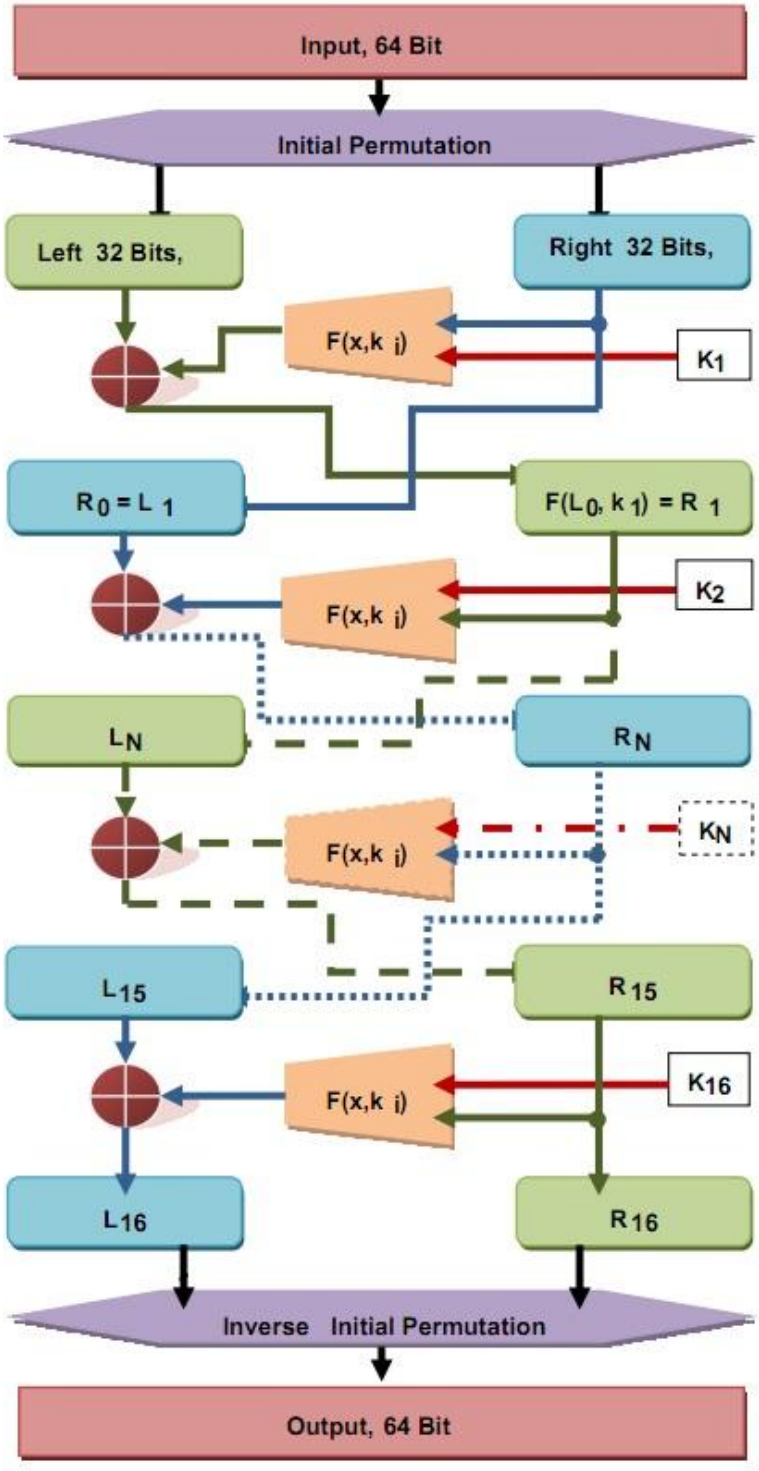
ANEXO A — Arquitetura do CI TUSB3410

Fonte: Texas Instruments [17]



ANEXO B — Estrutura do algoritmo de criptografia DES

Fonte:Texas Instruments [23]



ANEXO C — Estrutura do algoritmo de criptografia AES 128

Fonte: Texas Instruments [23]