**Observações a serem seguidas:**

* Escrever tudo na terceira pessoa do passado: O relatório é escrito pela empresa, descrevendo o que cada desenvolvedor fez;
* Cada justificativa deve começar por: “O colaborador...” ou “O foco do colaborador...”
* Resumir as etapas: Juntar as etapas com características em comum;
* Especificar as etapas e entregas de acordo com o plano de trabalho: usar as mesmas nomenclaturas;
* Considerar os critérios de avaliação:
  + Critério C1 - Existência de problema técnico-científico: objetivos que levem à resolução de um problema técnico científico
  + Execução sistemática, contextualizando as suas etapas com o escopo do projeto: objetiva identificar as atividades realizadas e se elas têm relação com o objetivo do projeto.
  + Critério C3 - Existência de atividades investigativas: identificar se houve a necessidade de testes e qual a natureza deles.
  + Critério C4 - Existência de um elemento de novidade tecnológica: acréscimo de conhecimentos ou práticas ao acervo tecnológico existente (novos conhecimentos, materiais, produtos, processos, etc. ou, pelo menos, aperfeiçoamentos significativos nesses materiais, produtos, processos, etc.).

**Macro Atividades**

**1. Atividade:** Preparação (Planejamento, Estudo e Configuração dos Ambientes de Desenvolvimento)

**Período de Execução:** 01/12/2017 a 31/12/2017

**MARIANA LINS**

O foco da colaboradora nesta atividade foi o entendimento do escopo do projeto, objetivos, metodologias e processos a serem utilizados e familiarização com as tecnologias e ferramentas utilizadas nas atividades subsequentes.

A colaboradora, em conjunto com os outros desenvolvedores e coordenadores, fizeram uma análise das solicitações e especificidades solicitadas no plano de projeto. Chegaram à conclusão que os principais objetivos a serem alcançados no final do projeto eram: o estudo e documentação no WIKI da placa Arduino M0 Pro, dos módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42, da IDE Atmel Studio e Framework ASF (Advanced Software Framework), da IDE e framework Arduino e a implementação e documentação do conjunto de bibliotecas em linguagem de programação C (para Atmel Studio) e C++ (para Arduino) para os módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42 e, por fim, das aplicações de exemplo para as plataformas Atmel Studio e Arduino.

A colaboradora participou também de reunião para definição de processos e ferramentas de gestão. A partir de experiências prévias bem sucedidas em projetos anteriores, decidiram utilizar o processo ágil Scrum, utilizando a ferramenta de gestão de tarefas Trello como ferramenta de Kanban. E para gestão de risco, decidiram utilizar o procedimento de mitigação por antecipação, colocando como prioridades as atividades que identificarem algum tipo de risco.

Após divisão de tarefas, a colaboradora configurou a ferramenta WIKI chamada MKDocs. Essa ferramenta foi sugerida no plano de projeto pelo cliente. Trata-se de uma ferramenta em Python que pode ser executada localmente, em ambiente Linux e Windows, com definição de formatação de conteúdos utilizando a linguagem Mark Down. Para melhor aparência e responsividade, a colaboradora instalou o plugin MKDocs-Material. Por fim a colaboradora fez um documento de readme.txt descrevendo o procedimento de instalação e atualização do Wiki para que outras pessoas pudessem configurar, instalar e inserir conteúdos na plataforma Wiki.

Na sequência a colaboradora fez uma análise do framework ASF (Advanced Software Framework) e do sistema operacional de tempo real FreeRTOS onde concluiu que o framework ASF é integrado com a IDE Atmel Studio e possui boa documentação e implementação de diversos drivers para a arquitetura de microcontroladores ATSAMD21, os utilizados na placa Arduino M0 Pro. Constatou também que o framework ASF inclui o sistema operacional FreeRTOS, o que facilita a sua integração no projeto. Um ponto negativo identificado por ela, é que não existe o projeto (chamado de board) pré configurado para a placa Arduino M0, de modo que foi necessário fazer uma engenharia reversa e análise de diagrama esquemático da placa Arduino M0 para identificar quais pinos do microcontrolador são utilizados para cada porta serial e ainda suas definições de frequência de cristal, pinos dos LEDs entre outras. O resultado desse trabalho foi um arquivo de definição da placa Arduino M0 para o Atmel Studio / ASF.

**PEDRO GRANVILLE**

O foco do colaborador nesta atividade foi o entendimento do escopo do projeto, objetivos, metodologias e processos a serem utilizados e familiarização com as tecnologias e ferramentas utilizadas nas atividades subsequentes.

O colaborador, em conjunto com os outros desenvolvedores e coordenadores, fizeram uma análise das solicitações e especificidades solicitadas no plano de projeto. Chegaram à conclusão que os principais objetivos a serem alcançados no final do projeto eram: o estudo e documentação no WIKI da placa Arduino M0 Pro, dos módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42, da IDE Atmel Studio e Framework ASF (Advanced Software Framework), da IDE e framework Arduino e a implementação e documentação do conjunto de bibliotecas em linguagem de programação C (para Atmel Studio) e C++ (para Arduino) para os módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42 e, por fim, das aplicações de exemplo para as plataformas Atmel Studio e Arduino.

O colaborador participou também de reunião para definição de processos e ferramentas de gestão. A partir de experiências prévias bem sucedidas em projetos anteriores, decidiram utilizar o processo ágil Scrum, utilizando a ferramenta de gestão de tarefas Trello como ferramenta de Kanban. E para gestão de risco, decidiram utilizar o procedimento de mitigação por antecipação, colocando como prioridades as atividades que identificarem algum tipo de risco.

Após divisão de tarefas, o colaborador configurou a ferramenta de gestão de tarefas Trello, criando os quadros “A Fazer”, “Fazendo”, “Bloqueadas”, “A Testar”, “A Documentar” e “Prontas” e adicionando os desenvolvedores e coordenadores envolvidos no projeto.

Em seguida o colaborador fez uma análise do framework Arduino onde concluiu que para utilizar a placa Arduino M0 Pro na plataforma Arduino era necessário uam configuração adicional desse modelo de placa na IDE. Para isso era necessário ir na opção Tools/Board/Board Manager e instalar o suporte a placas do tipo “Arduino SAMD boards”. Constatou também a necessidade de instalação do driver EDBG (mecanismos de gravação e debug utilizado na placa Arduino M0 Pro) e que no Windows essa instalação era feita automaticamente, no momento que se conecta a placa no computador.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nesta atividade foi o entendimento do escopo do projeto, objetivos, metodologias e processos a serem utilizados e familiarização com as tecnologias e ferramentas utilizadas nas atividades subsequentes.

O colaborador, em conjunto com os outros desenvolvedores e coordenador, fizeram uma análise das solicitações e especificidades solicitadas no plano de projeto. Chegaram à conclusão que os principais objetivos a serem alcançados no final do projeto eram: o estudo e documentação no WIKI da placa Arduino M0 Pro, dos módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42, da IDE Atmel Studio e Framework ASF (Advanced Software Framework), da IDE e framework Arduino e a implementação e documentação do conjunto de bibliotecas em linguagem de programação C (para Atmel Studio) e C++ (para Arduino) para os módulos Telit GSM HE910, Wifi GS2101M, e Bluetooth Bluemod+S42 e, por fim, das aplicações de exemplo para as plataformas Atmel Studio e Arduino.

O colaborador participou também de reunião para definição de processos e ferramentas de gestão. A partir de experiências prévias bem sucedidas em projetos anteriores, decidiram utilizar o processo ágil Scrum, utilizando a ferramenta de gestão de tarefas Trello como ferramenta de Kanban. E para gestão de risco, decidiram utilizar o procedimento de mitigação por antecipação, colocando como prioridades as atividades que identificarem algum tipo de risco.

Após divisão de tarefas, o colaborador configurou a ferramenta de controle de versões GIT, na plataforma gratuita de repositórios chamada BitBuket. O colaborador criou uma estrutura inicial de diretórios, para servir como primeiro commit.

Em seguida o colaborador fez uma análise inicial das ferramentas de software e kits de desenvolvimento de cada módulo da Telit, concluindo que os três módulos da Telit utilizam uma mesma ferramenta de testes chamada “Telit AT Controller”. Essa ferramenta disponibiliza um terminal de comunicação serial e diversos exemplos de sequências de comandos AT para testes das principais funcionalidades de cada módulo. Esses exemplos puderam ser utilizados como base para a criação das bibliotecas, além de possibilitar os testes iniciais dos módulos.

Por fim o colaborador fez uma pesquisa sobre os kits de desenvolvimento de cada módulo. Nessa pesquisa ele localizou as documentações de cada módulo e registrou os links no Wiki. Identificou também os kits de desenvolvimento dos módulos HE910, GS2101M e BlueMod+S42, que são respectivamente: Kit de desenvolvimento EVK2, Kit de Desenvolvimento GainSpan GS2100M EVB3 e Kit de desenvolvimento BlueEva+S42. As suas referências e documentações também foram inseridas no WIKI.

**2. Atividade:** Testes dos Módulos

**Período de Execução:** 01/12/2017 a 31/12/2017

**MARIANA LINS**

O foco da colaboradora nesta atividade foi efetuar testes funcionais no kit GainSpan GS2100M EVB3 e documentar os resultados no Wiki. Esse kit é composto por uma placa principal de suporte para o módulo Wifi GS2101M da Telit. Além da placa principal, inclui também fonte de alimentação e manual de utilização.

Para os testes foi necessário disponibilização de uma rede Wifi com criptografia do tipo WPA2 e utilização do aplicativo Telit AT Controller, aplicativo que detecta os módulos, quando conectados na porta USB no PC, e possibilita enviar e receber comandos AT para eles. Várias sequências de comandos pré definidas estão disponíveis para testar as funcionalidades do módulo.

Inicialmente a colaboradora se deparou com dificuldades para comunicação com o módulo. A primeira tentativa efetuada foi seguir o procedimento de atualização de firmware do módulo, especificado em uma documentação online, entretanto esse procedimento não resolveu o problema. Após alguns testes seguintes, ele identificou que a configuração da porta serial virtual (criada sobre a porta USB de comunicação com o kit), estava funcionando com um baudrate diferente do que estava descrito na documentação da Telit. A documentação indicava um baudrate de 115200bps entretanto, nos testes, foi detectado que o baudrate que funcionada era o de 9600bps.

Na sequência, a colaboradora identificou os pinos de TX e RX da porta UART do módulo no kit de desenvolvimento e fez a integração com a placa Arduino M0 Pro, testando o envio e recebimento de comandos AT feitos, dessa vez, entre o Arduino e o kit de desenvolvimento.

Por fim a colaboradora atualizou a documentação do Wiki com os resultados e informações úteis desse teste.

**PEDRO GRANVILLE**

O foco do colaborador nesta atividade foi efetuar testes funcionais no kit de desenvolvimento EVK2 e documentar os resultados no Wiki. Esse kit é composto por uma placa principal de suporte para o módulo GS HE910 da Telit. Além da placa principal, o kit inclui uma placa auxiliar, fonte de alimentação, antenas GSM e GPS, Chip SimCard da Telit para testes e manual de utilização.

Antes de iniciar os testes propriamente ditos, o colaborador tentou ativar o chip simcard disponibilizado junto com o kit, seguindo o passo a passo da documentação, entretanto link indicado para efetuar essa ativação não existia. Após algumas tentativas sem sucesso, o colaborador, em conjunto com os coordenadores, decidiram adquirir um chip simcard por conta própria para a execução dos testes.

Após estar com o chip em mãos e leitura da documentação do kit de desenvolvimento e do módulo, o colaborador efetuou os testes funcionais com sucesso, utilizando o aplicativo Telit AT Controller (aplicativo que detecta os módulos, quando conectados na porta USB no PC, e possibilita enviar e receber comandos AT para eles). Várias sequencias de comandos pré definidas estão disponíveis para testar as funcionalidades do módulo. Nesse teste o colaborador conectou na rede GSM/3G e, em seguida, fez uma conexão em um servidor HTTP, com sucesso.

Em seguida, o colaborador identificou os pinos de TX e RX da porta UART do kit de desenvolvimento e fez a integração com a placa Arduino M0 Pro, testando o envio e recebimento de comandos AT entre o Arduino e o kit, com sucesso.

Após considerar validado, o colaborador atualizou a documentação do Wiki com os resultados e informações úteis obtidos nesse teste.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nesta atividade foi efetuar testes funcionais no kit de desenvolvimento BlueEva e documentar os resultados no Wiki. Esse kit é composto por uma placa principal de suporte para o módulo Bluetooth BlueMod+S42 da Telit. Além da placa principal, o kit inclui uma placa auxiliar com antena, bateria, fonte de alimentação e manual de utilização.

Inicialmente o colaborador estudou a documentação do kit de desenvolvimento e montou o ambiente de testes composto pela placa devidamente montada e alimentada, e o software de testes da Telit (Telit AT Controller). Após inúmeras tentativas frustradas, o colaborador decidiu fazer contato por email com a equipe de suporte da Telit, comunicando a dificuldade e descrevendo os passos seguidos por ele. Após algumas interações com a equipe da Telit, fomos informados que o kit BlueEva enviado para nós estava com uma versão de firmware que não possibilitava a integração através de comandos AT e que eles iriam nos enviar um novo kit com a versão de firmware correta. O colaborado, então organizou a devolução dos kits que foram recebidos e acompanhou o recebimento dos novos kits de desenvolvimento.

Após a chegada nos novos kits, o colaborador efetuou os mesmos procedimentos de testes descritos na documentação e obteve sucesso, validando a funcionalidade correta do mesmo.

Em seguida, o colaborador identificou os pinos de TX e RX da porta UART do kit BlueEva com a placa Arduino M0 Pro, testando o envio e recebimento de comandos AT entre o Arduino e o kit, com sucesso.

Por fim o colaborador documentou no Wiki os procedimentos e resultados dos testes.

**3. Atividade:** Implementação de Firmwares Base e Bibliotecas de Comandos AT

**Período de Execução:** 01/01/2018 a 31/01/2018

**MARIANA LINS**

Nesta atividade a colaboradora teve como foco criar o projeto de firmware para a plataforma Atmel Studio, a ser usado como base para a implementação dos exemplos de aplicações, incluindo a integração com o sistema operacional FreeRTOS e implementação das bibliotecas de comandos AT.

Inicialmente a colaboradora criou a estrutura de diretórios, organizando os códigos fontes comuns (bibliotecas genéricas, de comandos AT e de comunicação com os módulos), em um subdiretório chamado de “libs”, e os exemplos de aplicações, sendo compostos pelos projetos propriamente ditos da IDE Atmel Studio, localizados no subdiretório chamado “examples”. A colaboradora sub dividiu ainda esses exemplos em outra camada de subdiretórios, segmentando esses exemplos por módulos (he910, gs2101m e bluemods42).

Para a criação do projeto base no Atmel Studio, a colaboradora precisou criar a “board” do Arduino m0 Pro. No Atmel Studio o termo “board” se refere às definições específicas de uma placa como, por exemplo, pinos de GPIO que estão ligados os LEDs ou quais portas seriais do microcontrolador serão utilizadas para cada finalidade. Como resultado dessa tarefa, a colaboradora criou um arquivo de header chamado conf\_ArduinoM0Pro\_board.h contendo essas definições da placa.

Uma vez criado o projeto no Atmel Studio com sucesso, a colaboradora se deparou com o desafio de fazer funcionar um firmware de exemplo básico, ligando e desligando os LEDs e escrevendo dados na porta serial da placa Arduino. Primeiro a colaboradora teve problemas para gravar o firmware na placa Arduino, através da plataforma Atmel Studio. Após pesquisas e testes, a colaboradora identificou que um fusebit de proteção de gravação da memória flash era ativado, quando gravado pela IDE do Arduino. A colaboradora precisou alterar esse fusebit manualmente para o seu valor de fábrica para resolver o projeto. O segundo desafio foi configurar o mecanismo de clock do microcontrolador, para utilizar a fonte de clock correta e com a frequência correta, tarefa essa que também foi executada com sucesso pela colaboradora, conseguindo, por fim, executar o firmware de teste básico.

Na sequência a colaboradora criou as bibliotecas das portas seriais (uma para comunicação com o módulo e a segunda para comunicação com a aplicação) e leds para sinalização, utilizando os drivers do framework ASF. Em seguida a colaboradora fez a integração do FreeRTOS com o projeto, criando uma task simples, usando as bibliotecas recém implementadas.

Por se tratar de um sistema operacional específico para microcontroladores, contendo características e metodologias de desenvolvimento específicas, foi necessário fazer um estudo profundo sobre o FreeRTOS para aprender os seus conceitos e mecanismos de aplicação.

O FreeRTOS oferece recursos como gerenciamento de memória, podendo utilizar alocação dinâmica ou estática, definido pelo usuário, implementações de filas, multi tarefas, controle de comunicação entre tarefas, utilizando mecanismos de mutexes e semáforos, além de muitos outros recursos e bibliotecas.

Os principais desafios ultrapassados pela colaboradora nessa tarefa foi a configuração do FreeRTOS para utilização no projeto (configuração de clocks e base de tempo, ajustes de tamanho de memória de Heap e Stack, definição das temporizações para cada task e o gerenciamento de troca de dados entre as tasks).

Após toda a base do firmware estar funcionando corretamente, a colaboradora iniciou a implementação da biblioteca de comandos AT. Foi criada uma biblioteca genérica de comandos AT assíncrona, tratando o envio de comandos e o recebimento por call-back, podendo ser utilizada em qualquer outro projeto em linguagem de programação C, diretamente.

Por fim a colaboradora documentou no WIKI os conhecimentos adquiridos e as especificações do projeto e das bibliotecas criadas nessa atividade.

**PEDRO GRANVILLE**

Nesta atividade o colaborador teve como foco criar o projeto de firmware para a plataforma Arduino, a ser usado como base para a implementação dos exemplos de aplicações, incluindo a implementação das bibliotecas de comandos AT.

A IDE Arduino possibilita integrar em sua estrutura de menus tanto bibliotecas quanto exemplos de aplicações, torando a utilização das bibliotecas e execução dos exemplos bastante intuitivos pelos usuários. Para utilizar esse mecanismo nas implementações para Arduino, o colaborador fez uma pesquisa de algumas bibliotecas de código Aberto para Arduino, contendo exemplos integrados com a IDE. Analisando as estruturas desses projetos, ele identificou o procedimento a ser feito para criar exemplos e bibliotecas integradas com a IDE. Para isso ele criou a estrutura de projeto contendo os subdiretórios doc (que contém as documentações da biblioteca), examples (que contém as estruturas de projetos e arquivos .ino que implementam cada exemplo) e os arquivos .h e .cpp: implementações das bibliotecas, na raiz da estrutura de diretórios. Esse procedimento foi documentado pelo colaborador no WIKI, com textos descritivos e imagens ilustrando os procedimentos de configuração.

Na sequência, o colaborador criou a biblioteca de comandos AT genérica para a plataforma Arduino, já dentro da estrutura do projeto para o módulo HE910. Ele criou também um projeto de teste para a primeira implementação que evoluiu posteriormente para se transformar no primeiro exemplo de aplicação. Os arquivos desenvolvidos foram arduino/Telit\_GSM/AT.cpp (Implementação dos métodos da biblioteca de comandos AT para arduino), e arduino/Telit\_GSM/AT.h (Definição da classe da biblioteca de comandos AT para arduino).

**ROOSEVELT VINÍCIUS**

Nessa atividade, o colaborador deu apoio aos outros desenvolvedores e atuou na qualidade de software.

Inicialmente ele identificou e testou as sequências de comandos AT necessárias para a conexão com a rede GSM/GPRS para o módulo HE910 e para conexão com a rede WIFI para o módulo GS2101M. Em seguida o colaborador documentou esses dados no Wiki.

Na medida que os outros desenvolvedores foram criando versões funcionais das implementações da biblioteca de comandos AT, o colaborador efetuou testes unitários e funcionais, validando cada função (e método) da biblioteca de comandos AT e criando rotinas de testes para verificar se o resultado obtido era igual ao resultado esperado. Primeiro efetuou os testes para a plataforma Arduino, que foi a primeira a ter a uma versão funcional da biblioteca e, posteriormente, para a plataforma Atmel Studio. Os problemas encontrados foram inseridos no Trello e identificados como Bugs a serem corrigidos pelos desenvolvedores. O colaborador executou os testes e acompanhou os retrabalhos até a correção de todos os bugs detectados.

**4. Atividade:** Implementação de Sistemas Logger (Modem Wifi, GSM e Bluetooth)

**Período de Execução:** 01/02/2018 a 28/02/2018

**MARIANA LINS**

O foco da colaboradora nessa atividade foi a criação dos exemplos de sistemas de logger baseados em modems para a plataforma Atmel Studio. Sendo

Antes de iniciar do desenvolvimento da aplicação propriamente dito, a colaboradora criou as versões iniciais das bibliotecas de cada módulo, que implementam a *task* de gerenciamento de comunicação e funções que proveem as funcionalidades dos módulos para a camada de aplicação, utilizando a plataforma de software Atmel Studio (ASF+FreeRTOS). Ela criou os códigos fontes gs2101m.c/.h (Implementação das funções que disponibilizam as funcionalidades do módulo WIFI GS2101M), he910.c/.h (Implementação das funções que disponibilizam as funcionalidades do módulo GSM HE910) e bluemods42.c/.h (Implementação das funções que disponibilizam as funcionalidades do módulo Bluetooth Bluemod+S42)**.** Durante essa atividade, ela implementou as funções de configuração dos modens e de conexão TCP através de socket cliente.

Em seguida a colaboradora implementou as aplicações de exemplo de Sistema de Logger Baseados em Modem. Ela implementou as versões para o modem GSM, WIFI e Bluetooth, sendo cada uma delas um projeto de exemplo diferente.

Como caso de teste, a colaboradora acessou a porta serial da placa Arduino utilizando o aplicativo de terminal Teraterm e na outra ponta, utilizou o aplicativo de conexão TCP/IP chamado Packet Sender no modo servidor TCP. As aplicações para os três modens ficavam em um loop tentando conectar no servidor TCP, em uma porta pré configurada e, ao conectar, abria um túnel possibilitando enviar dados entre o terminal Teraterm e o Packet Sender.

Foram feitos testes de quebra de conexão e reconexão, e envio de dados enquanto a conexão não estava estabelecida, sendo enviados no momento em que reestabelece a conexão (para isso a colaboradora utilizou os mecanismos de queues – filas - disponíveis no FreeRTOS).

**PEDRO GRANVILLE**

O foco do colaborador nessa atividade foi a criação dos exemplos de sistemas de logger baseados em modems para a plataforma Arduino.

O colaborador criou as versões iniciais das classes de cada módulo, que implementam os métodos que proveem as funcionalidades dos módulos para a camada de aplicação na linguagem C++. Ele criou os códigos fontes gs2101m.cpp/.h (que implementa os métodos das funcionalidades do módulo WIFI GS2101M), he910.cpp/.h (que implementa os métodos das funcionalidades do módulo GSM HE910) e bluemods42.cpp/.h (que implementa os métodos das funcionalidades do módulo Bluetooth Bluemod+S42)**.** Durante essa atividade, foram implementados os métodos de configuração dos modens e de conexão TCP através de socket client.

Em seguida o colaborador implementou as aplicações de exemplo de Sistema de Logger Baseados em Modem. Ele implementou as versões para o modem GSM, WIFI e Bluetooth, sendo cada uma delas um exemplo diferente no menu da IDE do Arduino.

Como caso de teste, o colaborador acessou a porta serial da placa Arduino utilizando o terminal disponível na própria IDE do Arduino e na outra ponta, utilizou o aplicativo de conexão TCP/IP chamado Packet Sender no modo servidor TCP. As aplicações para os três modens ficavam em um loop tentando conectar no servidor TCP, em uma porta pré configurada e, ao conectar, abria um túnel possibilitando enviar dados entre o terminal e o Packet Sender.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nessa atividade foi o desenvolvimento e aplicação de rotinas de testes, controle de qualidade e documentação dos aplicativos de exemplos de sistemas de Logger para modem, criados pelos outros desenvolvedores.

Antes de iniciar as atividades das aplicações de exemplo, o colaborador documentou todas as bibliotecas e classes dos três módulos criadas pelos outros desenvolvedores. O colaborador utilizou o padrão de documentação em código chamado Doxygen. Se trata de uma sintaxe específica de comentários que, utilizando a ferramenta Doxygen, é gerado um documento em html contendo todas as documentações das bibliotecas em formato html. Esses htmls resultantes foram integrados ao Wiki, podendo ser regerados sempre que houverem atualizações dos comentários dos códigos fontes.

Após conclusão das documentações das bibliotecas dos módulos, o colaborador especificou cada uma das aplicações de sistemas de Logger para modem no Wiki. Em seguida ele criou rotinas de testes unitários e testes funcionais, registrou os bugs detectados no Trello, acompanhou a implementação dos comandos de abertura de conexão, escrita, leitura e desconexão do protocolo TCP para o módulo HE910 e GS2101M. E por fim ele revisou as documentações das aplicações de exemplos desenvolvidas no Wiki, atualizando as suas características e particularidades após o resultado final estarem prontos.

**5. Atividade:** Implementação de Sistemas Geofence (GPS e LBS)

**Período de Execução:** 01/02/2018 a 28/02/2018

**MARIANA LINS**

O foco da colaboradora nessa atividade foi a criação dos exemplos de sistemas Geofence para a plataforma Atmel Studio. O termo Geofence se trata da delimitação de uma área geográfica através de coordenadas de latitudes e longitudes e a detecção de ultrapassagem desses limites pelo dispositivo monitorado.

Antes de iniciar do desenvolvimento da aplicação, a colaboradora implementou as funções que proveem as funcionalidades de localização por GPS e LBS do módulo HE910, utilizando a plataforma de software Atmel Studio (ASF+FreeRTOS). Ela atualizou os códigos fontes he910.c/.h com essas funcionalidades.

Em seguida a colaboradora implementou as aplicações de exemplos de Sistema de Geofence utilizando as tecnologias LBS e GPS, conforme solicitado no plano de projeto, sendo cada uma delas um projeto de exemplo diferente.

Como caso de teste, a colaboradora inseriu no código fonte as 4 coordenadas formando um polígono e, assim que a posição do módulo ultrapassa um dos limites desse polígono, um led acende, uma mensagem é enviada na porta serial da placa Arduino informando se entrou ou saiu da área pré definida.

**PEDRO GRANVILLE**

O foco do colaborador nessa atividade foi a criação dos exemplos de sistemas Geofence para a plataforma Arduino. O termo Geofence se trata da delimitação de uma área geográfica através de coordenadas de latitudes e longitudes e a detecção de ultrapassagem desses limites pelo dispositivo monitorado.

Antes de iniciar do desenvolvimento da aplicação, o colaborador implementou os métodos que proveem as funcionalidades de localização por GPS e LBS do módulo HE910, utilizando a plataforma de software Arduino. Ele atualizou os códigos fontes he910.cpp/.h com essas funcionalidades.

Em seguida o colaborador implementou as aplicações de exemplos de Sistema de Geofence utilizando as tecnologias LBS e GPS, conforme solicitado no plano de projeto, sendo cada uma delas um exemplo diferente dentro da estrutura de menu de exemplos da IDE Arduino.

Como caso de teste, o colaborador inseriu no código fonte as 4 coordenadas formando um polígono e, assim que a posição do módulo ultrapassa um dos limites desse polígono, um led acende, uma mensagem é enviada na porta serial da placa Arduino informando se entrou ou saiu da área pré definida.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nessa atividade foi o desenvolvimento e aplicação de rotinas de testes, controle de qualidade e documentação dos aplicativos de exemplos de sistemas de Geofence utilizando as tecnologias LBS e GPS, criados pelos outros desenvolvedores.

Antes de iniciar as atividades das aplicações de exemplo, o colaborador atualizou as documentações das bibliotecas e classes do módulo HE910, descrevendo as novas rotinas implementadas pelos outros desenvolvedores. O colaborador utilizou o padrão de documentação em código Doxygen e, em seguida, atualizou o Wiki com os htmls atualizados.

Após atualização das documentações da biblioteca do módulo HE910, o colaborador especificou cada uma das aplicações de sistemas de Geofence (LBS e GPS). Em seguida ele criou rotinas de testes unitários e testes funcionais, registrou os bugs detectados no Trello e, por fim revisou as documentações das aplicações de exemplos desenvolvidas no Wiki, atualizando as suas características e particularidades após o resultado final estarem prontos.

**6. Atividade:** Implementação de Sistemas de Publicação de variável em sistema MQTT

**Período de Execução:** 01/02/2018 a 28/02/2018

**MARIANA LINS**

O foco da colaboradora nessa atividade foi a criação dos exemplos de Publicação de variável em sistema MQTT para a plataforma Atmel Studio, para os módulos Telit GS2101M (Wifi) e HE910 (GSM). As aplicações leem uma fila de dados a serem enviados para o portal MQTT da Telit, em pares “nome”:“valor”.

Antes de iniciar do desenvolvimento da aplicação, a colaboradora implementou as funções que proveem as funcionalidades de MQTT para o módulo HE910 e integrou a biblioteca opensource Paho Embedded MQTT for C/C++ para utilização no exemplo do módulo WIFI GS2101M. Ela atualizou os códigos fontes he910.c/.h e gs2101m.c/.h com essas funcionalidades.

Em seguida a colaboradora implementou as aplicações de exemplos, conforme solicitadas no plano de projeto, sendo cada uma delas um projeto de exemplo diferente.

**PEDRO GRANVILLE**

O foco do colaborador nessa atividade foi a criação dos exemplos de Publicação de variável em sistema MQTT para a plataforma Arduino, para os módulos Telit GS2101M (Wifi) e HE910 (GSM). As aplicações leem uma fila de dados a serem enviados para o portal MQTT da Telit, em pares “nome”:“valor”.

Antes de iniciar do desenvolvimento da aplicação, o colaborador implementou as funções que proveem as funcionalidades de MQTT para o módulo HE910 e integrou a biblioteca opensource Paho Embedded MQTT for C/C++ para utilização no exemplo do módulo WIFI GS2101M. Ele atualizou os códigos fontes he910.cpp/.h e gs2101m.c/.h com essas funcionalidades.

Em seguida o colaborador implementou as aplicações de exemplos, conforme solicitadas no plano de projeto, sendo cada uma delas um exemplo diferente disponível no menu de exemplos da IDE Arduino.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nessa atividade foi o desenvolvimento e aplicação de rotinas de testes, controle de qualidade e documentação dos aplicativos de exemplos de sistemas de Publicação de Variáveis MQTT no Portal da Telit, para os módulos Wifi e GSM da Telit.

Antes de iniciar as atividades de testes e qualidade de software, o colaborador atualizou as documentações das bibliotecas e classes dos módulos HE910 e GS2101M, descrevendo as novas rotinas implementadas pelos outros desenvolvedores. O colaborador utilizou o padrão de documentação em código Doxygen e, em seguida, atualizou o Wiki com os htmls atualizados.

Após atualização das documentações da biblioteca dos módulos HE910 e GS2101M o colaborador especificou cada uma das aplicações dos sistemas de Publicação de Variáveis MQTT no Portal da Telit. Em seguida ele criou rotinas de testes unitários e testes funcionais, registrou os bugs detectados no Trello e, por fim, revisou as documentações das aplicações de exemplos desenvolvidas no Wiki, atualizando as suas características e particularidades após o resultado final estarem prontos.

**7. Atividade:** Implementação de Sistema de Comunicação por SMS

**Período de Execução:** 01/02/2018 a 28/02/2018

**MARIANA LINS**

O foco da colaboradora nessa atividade foi a criação do exemplo de Sistema de Comunicação Por SMS para a plataforma Atmel Studio, para o módulo Telit HE910 (GSM). A aplicação monitora continuamente a recepção de mensagens de texto SMS e, ao receber, interpreta o comando que vem no corpo da mensagem (ligar ou desligar um LED) e, em seguida, envia uma mensagem SMS de resposta para o número de origem da mensagem.

Antes de iniciar do desenvolvimento da aplicação, a colaboradora implementou as funções que proveem as funcionalidades de envio e recebimento de mensagens de texto SMS na biblioteca do módulo GSM HE910. Ela atualizou os códigos fontes he910.c/.h com essas funcionalidades. Em seguida a colaboradora implementou a aplicação de exemplo, conforme solicitada no plano de projeto.

**PEDRO GRANVILLE**

O foco do colaborador nessa atividade foi a criação do exemplo de Sistema de Comunicação Por SMS para a plataforma Arduino, para o módulo Telit HE910 (GSM). A aplicação monitora continuamente a recepção de mensagens de texto SMS e, ao receber, interpreta o comando que vem no corpo da mensagem (ligar ou desligar um LED) e, em seguida, envia uma mensagem SMS de resposta para o número de origem da mensagem.

Antes de iniciar do desenvolvimento da aplicação, o colaborador implementou as funções que proveem as funcionalidades de envio e recebimento de mensagens de texto SMS na biblioteca do módulo GSM HE910. Ele atualizou os códigos fontes he910.cpp/.h com essas funcionalidades. Em seguida o colaborador implementou a aplicação de exemplo, conforme solicitada no plano de projeto.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nessa atividade foi o desenvolvimento e aplicação de rotinas de testes, controle de qualidade e documentação da aplicação de exemplo do Sistema de Comunicação Por SMS, para o módulo GSM da Telit.

Antes de iniciar as atividades de testes e qualidade de software, o colaborador atualizou as documentações das bibliotecas e classes do módulo HE910, descrevendo as novas rotinas implementadas pelos outros desenvolvedores. O colaborador utilizou o padrão de documentação em código Doxygen e, em seguida, atualizou o Wiki com os htmls atualizados.

Após atualização das documentações da biblioteca do módulo HE910 o colaborador especificou a aplicação de exemplo. Em seguida criou as rotinas de testes unitários e testes funcionais, registrou os bugs detectados no Trello e, por fim, revisou as documentações da aplicação de exemplo desenvolvida no Wiki, atualizando as suas características e particularidades após o resultado final estar pronto.

**8. Atividade:** Implementação de Sistema de Porteiro Eletrônico

**Período de Execução:** 01/02/2018 a 28/02/2018

**MARIANA LINS**

O foco da colaboradora nessa atividade foi a criação do exemplo de Sistema de Porteiro Eletrônico para a plataforma Atmel Studio, para o módulo Telit HE910 (GSM). A aplicação efetua uma chamada de voz para um número de telefone pré definido quando o botão da placa Arduino for pressionado.

Antes de iniciar do desenvolvimento da aplicação, a colaboradora implementou as funções que proveem as funcionalidades de efetuar chamadas de voz na biblioteca do módulo GSM HE910. Ela atualizou os códigos fontes he910.c/.h com essas funcionalidades.

Em seguida a colaboradora implementou a aplicação de exemplo, conforme solicitada no plano de projeto.

**PEDRO GRANVILLE**

O foco do colaborador nessa atividade foi a criação do exemplo de Sistema de Porteiro Eletrônico para a plataforma Arduino, para o módulo Telit HE910 (GSM). A aplicação efetua uma chamada de voz para um número de telefone pré definido quando o botão da placa Arduino for pressionado.

Antes de iniciar do desenvolvimento da aplicação, o colaborador implementou as funções que proveem as funcionalidades de efetuar chamadas de voz na biblioteca do módulo GSM HE910. Ele atualizou os códigos fontes he910.c/.h com essas funcionalidades.

Em seguida o colaborador implementou a aplicação de exemplo, conforme solicitada no plano de projeto.

**ROOSEVELT VINÍCIUS**

O foco do colaborador nessa atividade foi o desenvolvimento e aplicação de rotinas de testes, controle de qualidade e documentação da aplicação de exemplo do Sistema de Porteiro Eletrônico, para o módulo GSM da Telit.

Antes de iniciar as atividades de testes e qualidade de software, o colaborador atualizou as documentações das bibliotecas e classes do módulo HE910, descrevendo as novas rotinas implementadas pelos outros desenvolvedores. O colaborador utilizou o padrão de documentação em código Doxygen e, em seguida, atualizou o Wiki com os htmls atualizados.

Após atualização das documentações da biblioteca do módulo HE910 o colaborador especificou a aplicação de exemplo. Em seguida criou as rotinas de testes unitários e testes funcionais, registrou os bugs detectados no Trello e, por fim, revisou as documentações da aplicação de exemplo desenvolvida no Wiki, atualizando as suas características e particularidades após o resultado final estar pronto.

-----------------------------------------------------------------------------------------------------------------------------------

Aqui estão os conteúdos criados anteriormente que estão sendo revisados e inseridos nos respectivos campos acima

**1. Atividade:** Planejamento e Preparação

**Período de Execução:** 01/12/2017 a 31/12/2017

**Horas na Atividade:** 11

**Justificativa de participação na atividade:**

Nos dedicamos inicialmente ao entendimento do escopo do projeto, objetivos, metodologias, processos a serem utilizados durante o desenvolvimento e familiarização com as tecnologias e ferramentas a serem utilizadas nas atividades subsequentes.

1. **Entendimento do Escopo e Objetivos**

Inicialmente foi feita uma análise em equipe das solicitações e especificidades solicitadas para serem desenvolvidas. Ficou entendido que os objetivos a serem alcançados no final do projeto são:

* 1. Estudo e documentação no WIKI da placa Arduino M0 Pro
  2. Estudo e documentação no WIKI do módulo Telit GSM HE910
  3. Estudo e documentação no WIKI do módulo Telit Wifi GS2101M
  4. Estudo e documentação no WIKI do módulo Telit Bluetooth Bluemod+S42
  5. Estudo e documentação no WIKI da IDE Atmel Studio e Framework ASF (Advanced Software Framework)
  6. Estudo e documentação no WIKI da IDE e framework Arduino
  7. Desenvolvimento e documentação no WIKI de conjunto de bibliotecas em linguagem de programação C utilizando a IDE Atmel Studio, Framework ASF e sistema operacional de tempo real FreeRTOS para os módulos da Telit HE910, GS2101M e BlueMod+S42
  8. Desenvolvimento e documentação no WIKI de bibliotecas em linguagem de programação C++ utilizando a IDE/Framwork Arduino para os módulos da Telit HE910, GS2101M e BlueMod+S42
  9. Desenvolvimento e documentação no WIKI de aplicações de exemplo para as plataformas Atmel Studio e Arduino:
     1. Sistema logger baseado em modem: Tasks contínuas (RTOS) / Loop infinito (no Arduino) que repassa o tráfego de uma das portas seriais para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Com a possibilidade de bufferizar os bytes que foram recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins devem ser manuais, através de constantes no código. O sistema deverá verificar que a conexão caiu (contexto com a APN ou o socket caíram) e tentar continuamente uma reconexão. Permitir também o fluxo de dados servidor→dispositivo, refletido na porta serial também. Basicamente um tunel de porta serial por rede celular e TCP.
     2. O mesmo sistema logger acima, baseado em WiFi, através dos comandos AT para sockets.
     3. O mesmo sistema logger acima, baseado em Bluetooth / comandos AT Terminal IO do módulo.
     4. Sistema logger de Bluetooth através de modem: coloca o módulo Bluetooth para ouvir beacons e broadcasts / advertisement próximos e repassa eles diretamente para um socket aberto no modem.
     5. Sistema logger de Bluetooth, baseado em Wifi.
     6. Sistema geofence para GPS: Determinar o pertencimento de uma latitude / longitude recebida através dos códigos NMEA emitidos automaticamente pelo módulo GPS, pela porta serial, com o interior polígono definido através de um array de coordenadas. Quando o fix, representado pela mensagem $GPGGA, estiver fora do polígono, enviar onda quadrada para um buzzer / falante e acender um LED.
     7. Sistema de geofence por LBS (#MONI). Utilização dos comandos AT#MONI e AT#AGPSSND para obtenção de posição por triangulação por rede celular, com análise de polígono considerando a grande imprecisão da posição obtida. Os comandos são simples de utilizar: AT#MONI=7 (Aguarda), AT#MONI, AT#AGPSSND (receberá lat/long). Devido à imprecisão, o status de ³dentro e fora do polígono² pode ser gradual, ao invés de binário / tudo ou nada.
     8. Publicação de variável em sistema MQTT, comandos AT de MQTT (método fácil) para modem: Criar task ou rotina (arduino) que lê uma fila de dados a serem enviados para o portal MQTT da Telit, em pares ³nome da da variável² e ³valor², e para cada item executa os comandos AT de post de variáveis por MQTT embutido nos módulos.
     9. Publicação de variável em sistema MQTT, comandos AT de Socket (método direto) para modem: Criar task ou rotina (arduino) que lê uma fila de dados a serem enviados para o portal MQTT da Telit, em pares ³nome da da variável² e ³valor², e para cada item efetua chamadas a uma biblioteca para o protocolo MQTT, se conectando ao servidor através dos comandos AT de socket do módulo.
     10. Publicação de variável em sistema MQTT, comandos AT de MQTT para o módulo WiFi.
     11. Publicação de variável em sistema MQTT, comandos AT de Socket para o módulo WiFi.
     12. Comunicação por SMS: task ou rotina que lê continuamente as mensagens SMS recebidas pelo módulo, lendo seu conteúdo e executando comandos conforme o texto enviado pela mensagem, com resposta por SMS.
     13. Sistema de atendimento de ligações com recebimento de comandos por DTMF e resposta/menu através de sons pré-gravados. Os comandos de detecção de DTMF possuem utilização simples e para tocar os sons basta chamar AT#APLAY=1,<diretório>,<nome\_do\_arquivo>. Ou AT#SAMR. Implementar menus de exemplo e em alguns deles o modem deverá enviar mensagens de SMS conforme o código apertado, enviar variáveis para o servidor MQTT conforme o código apertado.
     14. Sistema porteiro eletrônico: se um botão for pressionado, liga para um telefone predefinido.

1. **Definição da Metodologia de Desenvolvimento**

Utilizaremos o processo ágil Scrum como referência principal em relação a processo, utilizando a ferramenta de gestão de tarefas Trello utilizando como ferramenta de Kanban contendo os quadros “A Fazer”, “Fazendo”, “Bloqueadas”, “A Testar”, “A Documentar” e “Prontas”.

Colocaremos sempre como prioridades as atividades que possuem riscos técnicos, com o objetivo de sanamos eventuais problemas ou dificuldades no início da execução do projeto, antecipando a detecção do problema e sua resolução.

Os desenvolvedores farão os estudos das tecnologias em conjunto, de maneira horizontal, entretanto cada desenvolvedor será responsável por desenvolver e documentar um determinado escopo do projeto:

**Mariana Lins:** Desenvolvimento e documentação das bibliotecas na Plataforma Atmel Studio

**Pedro Granville:** Desenvolvimento e documentação das bibliotecas na Plataforma Arduino

**Roosevelt Vinicius:** Desenvolvimento e documentação das aplicações nas plataformas Atmel Studio e Arduino

E quando um dos desenvolvedores finalizar as suas atividades, irá ajudar nas atividades do outro que estiver com a maior quantidade de atividades a serem finalizadas.

Utilizaremos um repositório para controle de versão e não será necessário a utilização de branches, uma vez que cada desenvolvedor trabalhará em partes distintas dos códigos fontes e documentações. Os commits deverão ser feitos sempre que uma versão parcial ou final for concluída, sendo um intervalo máximo de 7 dias entre cada commit por um desenvolvedor.

1. **Definição, Instalação e Configuração das Ferramentas de Gestão**

Decidimos utilizar as seguintes ferramentas:

**Gestão de Conhecimento (WIKI):** Mkdocs + plugin Mkdocs-material

**Gestão de Tarefas:** Trello

**Controle de Versões:** Git utilizando a plataforma Bitbucket

Ficou decidido a seguinte divisão para configuração dessas ferramentas:

**Mariana Lins:** Criação e configuração do Mkdocs

**Pedro Granville:** Criação e configuração do Trello

**Roosevelt Vinicius:** Criação e configuração do Git no Bitbucket

1. **Análise das Tecnologias Envolvidas**
   1. Módulo Telit GSM HE910
      1. Conjunto de comandos ATs do módulo
      2. Conexão TCP cliente
      3. Sistema de Localização GPS
      4. Sistema de Localização LBS
      5. Protocolo MQTT
      6. Envio e Recebimento de SMS
      7. Lligação e Atendimento de Chamadas de Voz
      8. Detecção de tons DTMF
      9. Gravação e Reprodução de Arquivos de Audio
   2. Módulo Telit Wifi GS2101M
      1. Conjunto de comandos ATs do módulo
      2. Conexão TCP cliente
      3. MQTT
   3. Módulo Telit Bluetooth BlueMod+S42
      1. Bluetooth LE (Low Energy)
      2. Beacons e broadcasts / adverting
      3. RFCOMM
2. **Análise das Ferramentas de Desenvolvimento a serem Utilizadas**

**Mariana Lins**

Análise do framework ASF e sistema operacional de tempo real FreeRTOS

Foi feita a análise inicial do framework ASF e do sistema operacional de tempo real FreeRTOS onde cheguei nas seguintes conclusões:

O framework ASF é integrado com a IDE Atmel Studio e possui boa documentação e implementação de diversos drivers para a arquitetura de microcontroladores ATSAMD21, utilizada na placa Arduino M0 Pro. Os principais drivers que iremos utilizar serão os drivers de comunicação pela porta serial, driver de utilização de timers e GPIOs. Foi constatado ainda que o framework ASF já inclui também a integração com o sistema operacional FreeRTOS, o que facilitará a sua integração no projeto.

Um ponto negativo, é que não existe o projeto (chamado de board) pré configurado para a placa Arduino M0, de modo que será necessário fazer uma engenharia reversa e análise de diagrama esquemático da placa Arduino M0 para identificar quais pinos do microcontrolador são utilizados para cada porta serial e ainda suas definições de frequência de cristal, entre outras. O resultado desse trabalho será um arquivo de definição da placa Arduino M0 para o Atmel Studio / ASF.

**Pedro Granville**

Foi feita a análise inicial do framework Arduino onde cheguei nas seguintes conclusões:

Para a utilização da placa Arduino M0 Pro na plataforma Arduino é necessário a instalação adicional da placa na IDE. Para isso basta ir na opção Tools/Board/Board Manager e instalar a placa “Arduino SAMD boards”.

Também é necessária a instalação do driver EDBG (mecanismos de gravação e debug utilizado na placa Arduino M0 Pro) entretanto no Windows essa instalação é feita automaticamente, no momento que se conecta a placa no computador.

**Roosevelt Vinicius**

Foi feita a análise inicial das ferramentas de software e kits de desenvolvimento de cada módulo da Telit onde cheguei nas seguintes conclusões:

Os três módulos da Telit utilizam uma mesma ferramenta de testes chamada “Telit AT Controller”. Essa ferramenta disponibiliza um terminal de comunicação e diversos exemplos de sequências de comandos AT para teste das principais funcionalidades de cada módulo. Esses exemplos poderão ser utilizados como base para a criação das bibliotecas, além de possibilitarem os testes iniciais dos módulos.

Em seguida foi feito uma pesquisa dos kits de desenvolvimento de cada módulo. Nessa pesquisa foram identificadas as documentações e conhecidas as suas características. Os kits de desenvolvimento dos módulos HE910, GS2101M e BlueMod+S42 são respectivamente: Kit de desenvolvimento EVK2, Kit de Desenvolvimento GainSpan GS2100M EVB3 e Kit de desenvolvimento BlueEva+S42. As suas referências e documentações foram inseridas no WIKI.

**2. Atividade:** Estudo e Criação dos Projetos e Documentações Iniciais

**Período de Execução:** 01/12/2017 a 31/12/2017

**Horas na Atividade:** 44

**Justificativa de participação na atividade:**

**Mariana Lins**

O objetivo desta etapa é definir a estrutura e criar um projeto inicial para a plataforma Atmel Studio contendo os principais componentes de software que serão utilizados na implementação das bibliotecas e exemplos a serem desenvolvidos para esta plataforma. Para isso foram feitas as seguintes atividades:

**Proposta inicial de estrutura de projeto:**

Foi feita a especificação da estrutura de arquivos a ser adotada, organizando as bibliotecas, que serão comuns para todos os projetos e os projetos propriamente ditos, que conterão os arquivos específicos para cada um dos exemplos a serem desenvolvidos. Chegamos a seguinte estrutura:

* **asf:** Neste subdiretório conterão todos os projetos, bibliotecas e exemplos de aplicações para a plataforma Atmel Studio
  + **libs:** Neste subdiretório serão armazenados todos os códigos fontes compartilhados para todos os projetos de exemplos. Nele ficarão as bibliotecas dos módulos e implementações genéricas.
    - **inc:** Neste subdiretório ficarão os arquivos de cabeçalho (headers ou includes) compartilhados para todos os projetos de exemplos.
    - **src:** Neste subdiretório ficarão as implementações das bibliotecas dos módulos e implementações genéricas, comuns a todos os projetos de exemplos.
  + **examples:** Neste subdiretório ficarão as implementações de cada exemplo. E cada exemplo será composto por um projeto do Atmel Studio e seus códigos fontes específicos.
    - **he910:** Neste subdiretório estarão todos os projetos de exemplos para o módulo Telit GSM HE910
      * **tunnelTcp:** Estrutura interna do projeto e códigos fontes específicos
      * **geofenceGps:** Estrutura interna do projeto e códigos fontes específicos
      * **geofenceLbs:** Estrutura interna do projeto e códigos fontes específicos
      * **mqttHard:** Estrutura interna do projeto e códigos fontes específicos
      * **geofenceMqtt:** Estrutura interna do projeto e códigos fontes específicos
      * **comandosSms:** Estrutura interna do projeto e códigos fontes específicos
      * **geofenceSms:** Estrutura interna do projeto e códigos fontes específicos
      * **voiceMenuSms:** Estrutura interna do projeto e códigos fontes específicos
      * **EletronicPorter:** Estrutura interna do projeto e códigos fontes específicos
    - **gs2101m:** Neste subdiretório estarão todos os projetos de exemplos para o módulo Telit Wifi GS2101M
      * **tunnelTcp:** Estrutura interna do projeto e códigos fontes específicos
      * **mqttSoft:** Estrutura interna do projeto e códigos fontes específicos
    - **bluemod-s42:** Neste subdiretório estarão todos os projetos de exemplos para o módulo Telit Bluetooth Bluemod+S42
      * **tunnelTcp:** Estrutura interna do projeto e códigos fontes específicos
      * **beaconsTcp:** Estrutura interna do projeto e códigos fontes específicos

**Estudo e testes para criação do projeto Atmel Studio**

Foi criado com sucesso um projeto na IDE Atmel Studio para o mcu SAMD21G18 (o mcu utilizado na placa Arduino M0 Pro) com uma board customizada pelo usuário e inserido a versão mais atual do framework ASF.

**Estudo e testes para criação da board Arduino M0 Pro**

No Atmel Studio o termo “board” se refere às definições específicas de uma placa como, por exemplo, pinos de GPIO que estão ligados os LEDs ou quais portas seriais do microcontrolador serão utilizadas para cada finalidade.

Para a definição da board para a placa a ser utilizada, a Arduino M0 Pro, será necessário criar um header chamado conf\_ArduinoM0Pro\_board.h contendo essas definições da placa.

**Estudo e testes de utilização dos drivers ASF e integração com o FreeRTOS**

Em seguida foi feito um estudo inicial sobre o FreeRTOS e como integrá-lo a um projeto no Atmel Studio. Foi localizado uma documentação no próprio site do FreeRTOS com o passo a passo para isso no endereço: <http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42138-Getting-Started-with-FreeRTOS-on-SAM-D20-D21-R21-L21-L22_ApplicationNote_AT03664.pdf>

**Estudo do sistema operacional de tempo real FreeRTOS**

Um microcontrolador (MCU) é um único chip com um processador simples que pode ser encontrado em diversos dispositivos, como aparelhos, sensores, controladores de aptidão física, automação industrial e automóveis. Muitos desses pequenos dispositivos podem se beneficiar com a conexão à nuvem ou com a conexão local a outros dispositivos. Por exemplo, medidores de consumo elétrico inteligentes precisam se conectar à nuvem para relatar o uso e sistemas de segurança predial precisam se comunicar localmente para destrancar uma porta quando você usa o crachá para entrar. Os microcontroladores têm poder computacional e capacidade de memória limitados e normalmente executam tarefas funcionais simples. Muitas vezes, os microcontroladores executam sistemas operacionais que não têm funcionalidades incorporadas para conexão a redes locais ou à nuvem, o que é um desafio para os aplicativos de IoT. O FreeRTOS ajuda a resolver esse problema oferecendo o sistema operacional central (para execução no dispositivo) e as bibliotecas de software que facilitam a conexão segura com a nuvem (ou com outros dispositivos) para que você possa coletar dados deles para os aplicativos de IoT e executar ações.

Por se tratar de um sistema operacional específico para microcontroladores, contendo características e metodologias de desenvolvimento específicas, foi necessário fazer um estudo profundo sobre o FreeRTOS para aprender os seus conceitos e mecanismos de aplicação.

O FreeRTOS oferece recursos como gerenciamento de memória, podendo utilizar alocação dinâmica ou estática, definido pelo usuário, implementações de filas, multi tarefas, controle de comunicação entre tarefas, utilizando mecanismos de mutexes e semáforos, além de muitos outros recursos e bibliotecas.

Os principais desafios identificados inicialmente serão:

1. Configuração do FreeRTOS para utilização no projeto
   1. Configuração de clocks e base de tempo
   2. Ajustes de tamanho de memória de Heap e Stack
   3. Definição das temporizações para cada task
2. Gerenciamento de troca de dados entre as tasks
   1. Como teremos pelo menos duas tasks em cada projeto, uma responsável pela aplicação principal, e a outra pelo gerenciamento da comunicação com o módulo (biblioteca), será necessário utilizar tanto filas quanto mutexes para fazer esse gerenciamento.

Para testes foi criado um projeto de exemplo utilizando o FreeRTOS com uma task básica.

**Documentação**

Por fim foi documentado no WIKI os conhecimentos adquiridos nesses testes e estudos.

**Pedro Granville**

**Estudo de Procedimento para Criação de Bibliotecas e Exemplos na IDE Arduino**

Foi feito uma pesquisa de algumas bibliotecas contendo exemplos integrados com a IDE arduino e, a partir de suas estruturas, foi identificado o procedimento a ser feito para essa integração.

A IDE Arduino possibilita integrar em sua estrutura interna tanto a biblioteca quanto os exemplos desenvolvidos, criando uma estrutura pré definida pela IDE Arduino, da seguinte forma:

* diretório base da biblioteca
  + **doc:** subdiretório contendo arquivos textos contendo as documentações da biblioteca.
  + **examples:** subdiretório que contém as estruturas de projetos e arquivos .ino que implementam cada exemplo
  + **arquivos .h e .cpp**: implementações da biblioteca

**Criação de Estrutura Inicial**

Foi definida e criada a estrutura inicial para as bibliotecas e exemplos da plataforma Arduino, de forma a serem integradas com a IDE.

**Documentação**

Esse procedimento foi documentado no WIKI, com textos descritivos e imagens ilustrando os procedimentos de configuração.

**Roosevelt Vinicius**

Para entendimento com mais detalhes das funcionalidades e características de cada módulo, foi feito um estudo sobre cada um dos módulos a serem utilizados no projeto:

1. **HE910**

O módulo HE910 faz parte da família de módulos GSM da Telit xE910 destinados para aplicações automotivas e telemáticas 3G e 2G. Disponibiliza recursos avançados como:

* 1. Efetua e recebe chamadas de voz
  2. Envia e recebe mensagens de texto SMS
  3. Envia e recebe tons DTMF
  4. Localização GPS
  5. Localização LBS
  6. Pilha TCP/IP com conexões Socket TCP e UDP, podendo ter até 16 conexões simultaneas
  7. Protocolos HTTP, MQTT e outros.

1. **GS2101M**

A série GS2101M é um módulo Wi-Fi (IEEE 802.11 b / g / n) totalmente certificado e seguro que oferece integração rápida e econômica da conectividade Wi-Fi aos projetos de produtos. A série está disponível com uma antena de rastreio PCB opcional ou conector U.FL.

**Características:**

* 3 ADCs sigma-delta de 16 bits, ideais para dispositivos inteligentes de medição e energia
* 18 x 25 x 2,7 mm 40 pinos
* Medição de alta resolução e medição de energia
* Modo sem host (sem MCU externo), executando Wi-Fi completo e pilha de rede TCP/IP
* Modo host, a biblioteca de comandos AT mais abrangente conectando-se a qualquer microcontrolador host de 8/16/32 bits usando interfaces UART, SPI ou SDIO
* Fácil provisionamento de smartphones com AP limitado ou Wi-Fi Protected Set-up (WPS)
* Baixo consumo com recursos de gerenciamento de energia: 2,7V - 3,6V, 3,3V nominal

1. **BlueMod+S42**

O BlueMod+S42 é um módulo plug-and-play Bluetooth 4.2 Low Energy, de modo único, qualificado, de alta performance, com Bluetooth 4.2. Ele apresenta baixo consumo de energia e flexibilidade para projetistas como o perfil I/O de terminal para casos de uso de substituição de cabo serial, bem como a maioria dos perfis Bluetooth baseados em GATT por meio de uma implementação genérica do GATT. Parte da família BlueMod + Sx, o BlueMod + S42 é intercambiável com as gerações anteriores sem precisar de novos spins de hardware ou software. Embora seja um módulo de baixo consumo de energia de modo único, ele oferece uma alta taxa de transferência de até 470kbit/s usando o SPP exclusivo da Telit, como o serviço Terminal I/O.

**Características:**

* Plataforma IOT autônoma ou aplicativos hospedados por Bluetooth
* Módulo qualificado Bluetooth v4.2
* RED, FCC, IC, KC, MIC, RCM e Certificação Anatel
* Poderoso Cortex M4F MCU
* Projetado para aplicações de baixa potência
* Melhor do seu desempenho de intervalo de classe
* Melhor desempenho de rendimento da sua classe
* Cliente genérico GATT ou implementação de servidor
* Comunicações ponto a ponto ou ponto a multiponto
* Perfil de E/S de terminal (semelhante a SPP) para transferência de dados transparente rápida e fácil
* Perfil de E/S de automação para transferência de informações de status digital e analógico
* I/O de terminal e código de amostra de E/S de automação para iOS e Android
* Suporte NFC para emparelhamento simples seguro
* LE Secure Connection com criptografia AES de 128 bits

**3. Atividade:** Testes dos Módulos e Montagem dos Ambientes de Desenvolvimento - Parte I

**Período de Execução:** 01/12/2017 a 31/12/2017

**Horas na Atividade:** 20

**Justificativa de participação na atividade:**

**MARIANA LINS**

Recebemos 4 placas Arduino M0 Pro, 2 kits de desenvolvimento EVK2, do módulo HE910, 2 kits de desenvolvimento GainSpan GS2100M EVB3, do módulo GS2101M e 2 kits de desenvolvimento BlueEva, do módulo Bluemod+S42. Em comum acordo, decidimos que Pedro Granville iria testar o kit EVK2, eu iria testar o kit GainSpan GS2100M EVB3 e Roosevelt Vinicius o kit BlueEva.

**Testes do kit GainSpan GS2100M EVB3:**

Após os testes chegamos ao seguinte procedimento:

**Configuração do Ambiente de Hardware:**

Os procedimentos de setup inicial serão listados a seguir:

1 - Conectar a placa EVB3 através de seu conector microUSB (USB0) na porta USB do computador (a alimentação da placa ocorre pela USB do computador).

2 - Instalar o drive FTDI.

3 - Instalar o software serial TeraTerm VT termial.

4 - Executar o TeraTerm e configurar a porta serial como nas imagens abaixo:

**Obs. 1:** Os primeiros testes realizados com o kit EVB3 só funcionaram com o baudrate 460800 para a versão 3.5.0 do firmware de avaliação (AT+VER=?? -> comando que retorna a versão de firmware instalado na placa).

**Obs. 2:** O envio de comandos AT também podem ser realizados através do software AT Controller disponível na área de downloads no site da Telit.

Atualização de Firmware

O procedimento de atualização de firmware foi realizado seguindo a documentação técnica: GS2K Module Programming User Guide.

1 - Acessar o site http://www.gainspan.com.br e realizar o cadastro para ter acesso a portal de suporte.

2 - Clique em SDK Buikder e selecione o módulo GS2000.

Obs. 1: Clique em next na próxima tela.

3 - Na tela apresentada abaixo, foi selecionada opção para criação do firmware de avaliação da EVB3.

Obs: 1: Após clicar em next nesta tela, uma versão do firmware, com as configurações selecionadas, será criada. Em seguida ao clicar na aba Build History, o projeto do firmware, com a ferramenta de atualização e documentações, estará disponível para download em alguns minutos.

4 - Configure a placa para o modo de program na chave (vermelha), conecte a placa pela usb0 no computador e ligue a chave de alimentação da placa.

5 - Na pasta Tools está disponível a ferramenta de atualização de firmware GS\_programming\_tool.

Obs. 1: Selecione os parâmetros de conexão e clique em check connection para verificar se existe comunicação com a placa. Caso o software não consiga se comunicar com a placa basta selecionar o baudrate adequado, com o jumper J13 presente o baudrate é igua a 921600 e sem o jumper o baudrate é 115200.

Obs. 2: Em seguida, nas opções Super Block e Current FW Version, selecione os dois arquivos binários de atualização dentro da pasta Embedded\s2w e clique em Erase and Program.

Integração com Arduino M0 Pro

A Conexão da EVB3 com o arduino deve ser:

**Pino Arduino M0 Pro Pino EVB3**

GND GND

3.3V VIN\_3V3

10 GPIO0/UART0\_RX\_M

11 GPIO1/UART0\_TX\_M

09 EXT\_RTC\_RESET\_N

**PEDRO GRANVILLE**

Recebemos 4 placas Arduino M0 Pro, 2 kits de desenvolvimento EVK2, do módulo HE910, 2 kits de desenvolvimento GainSpan GS2100M EVB3, do módulo GS2101M e 2 kits de desenvolvimento BlueEva, do módulo Bluemod+S42. Em comum acordo, decidimos que eu iria testar o kit EVK2, Mariana Lins iria testar o kit GainSpan GS2100M EVB3 e Roosevelt Vinicius o kit BlueEva.

**Testes do kit GainSpan GS2100M EVB3:**

Após os testes chegamos ao seguinte procedimento:

**Configuração do Ambiente de Hardware:**

O ambiente de hardware utilizado para os exemplos consiste na integração da placa Arduino M0 Pro com o Kit de Desenvolvimento EVK2.

Os procedimentos de setup inicial serão listados a seguir:

1 - Baixar e instalar os drives USB's do link: Drivers..

2 - Conectar a placa CS1467C através de seu conector miniUSB na porta USB do computador, conectar a fonte de alimentação na placa base (EVK2) e pressionar o botão de on por 5s.

3 - Para testar a comunicação com o módulo através de comandos AT, deve-se instalar o software serial Telit AT Controller, disponível no mesmo link usado anteriormente pra baixar os drivers.

Configurada a conexão, basta clicar então no botão de "Connect", que é o botão da esquerda no meio da tela. Dando tudo certo e com o módulo conectado, são exibidas uma série de informações tais como IMEI, Fabricante, Modelo e versão de Firmware no centro da tela do programa, indicando que o mesmo está comunicando e recebendo dados do módulo.

Integração com Arduino M0 Pro

A Conexão da EVK2 com o arduino deve ser de acordo com a seguinte pinagem:

Pino Arduino M0 Pro Pino EVK2

GND GND

10 C104/RXD

11 C103/TXD

09 RESET

As descrições dos pinos utilizados na placa EVK2 podem ser observadas documento: EVK2 User Guide

**ROOSEVELT VINICIUS**

Recebemos 4 placas Arduino M0 Pro, 2 kits de desenvolvimento EVK2, do módulo HE910, 2 kits de desenvolvimento GainSpan GS2100M EVB3, do módulo GS2101M e 2 kits de desenvolvimento BlueEva, do módulo Bluemod+S42. Em comum acordo, decidimos que Pedro Granville iria testar o kit EVK2, Mariana Lins iria testar o kit GainSpan GS2100M EVB3 e eu o kit BlueEva.

**Testes do Kit BlueEva+S42:**

Não tivemos sucesso com os testes com este kit. Após várias tentativas decidimos fazer contato com a equipe da Telit através do seguinte email:

***Subject:*** *Suporte técnico para o módulo BluDev+S42*

*Oi, pessoal.*

*Boa tarde.*

*Vocês poderiam nos indicar um contato para suporte técnico relativo ao módulo BluDev+S42?*

*Segue abaixo uma breve descrição do problema. Em resumo, não estamos conseguindo nenhuma resposta aos comandos AT enviados, mesmo seguindo toda a documentação encontrada. Conseguimos nos comunicar com comandos AT com os outros módulos, menos esse. Testamos em todos os baudrates e configurações possíveis, sem sucesso. O estranho é que é o mesmo comportamento nos dois módulos. Então ainda pode ser que seja algum procedimento errado nosso.*

*Precisamos também liberar os chips GSM que vieram com o kit GSM. Contudo, o link que é passado na documentação para desbloquear não existe. Fizemos contato com o canal de suporte da Telit e estamos aguardando resposta. Terias algum canal mais rápido para nos ajudar?*

*Problema*

*As duas placas recebidas modelo: BlueDev+S42, não respondem aos comandos AT’s enviados pelo software AT controller, disponibilizado na área de download do site da Telit, nem pelo software terminal TeraTerm. Também não respondem ao procedimento de atualização de firmware.*

*Procedimento adotado*

*Foram seguidos detalhadamente os procedimentos listados no documento disponibilizado no site do fabricante do módulo (Telit), encontrado no link abaixo. Foram observados, desde a posição dos jumpers, como também a instalação dos drivers, softwares e configuração da porta serial para o devido funcionamento da placa.*

*https://www.telit.com/wp-content/uploads/2018/01/BlueEvaS42\_Central\_User\_Guide\_r1.pdf*

*https://www.telit.com/wp-content/uploads/2016/08/BlueMod\_S42\_Software\_User\_Guide\_r1.pdf*

Após esse contato tivemos a seguinte resposta da equipe da Telit:

*Subject: RE: Suporte técnico para o módulo BluDev+S42 | Case Number: 00027852*

*Boa tarde,*

*Verifiquei esse P/N e este é um kit de desenvolvimento voltado para a programação em C interna do módulo. Desta forma ele não vem com um firmware próprio que contempla os comandos AT. Estou verificando internamente o que pode ser feito nesse caso e já te retorno na sequência.*

*Quanto a ativação do simcard que vêm junto com o kit estou montando algumas instruções de ativação e já passo para você. Você poderia me passar o ICCID dos simcards? Este número geralmente esta impresso no simcard ou pode ser obtido pelo comando AT: AT+CCID*

*Att,*

*Jonatas Pazin*

*AMERICAS Technical Support*

*Telit Application Engineering*

Dessa forma ficou constatado que de fato havia um problema nos kits BlueEva enviado para nós. O problema estava no tipo de firmware carregados nos módulso dos kits enviados para nós e ficamos no aguardo do envio dos kits com o firmware correto.

**4. Atividade:** Implementação de Biblioteca de Comandos AT

**Período de Execução:** 01/01/2018 a 31/01/2018

**Horas na Atividade:** 38

**Justificativa de participação na atividade:**

**MARIANA LINS**

Implementação de Biblioteca de Comandos AT para a plaforma Atmel Studio

Foi feita a implementação da biblioteca de comunicação com as portas seriais do MCU Microchip SAMD21 (utilizado na placa Arduino M0 Pro) e de gerenciamento dos comandos AT para a plataforma de software Atmel Studio (ASF+FreeRTOS), necessários para que as bibliotecas a serem implementadas pudessem se comunicar com cada um dos módulos Telit.

Foram criados os códigos fontes:

**asf/libs/src/at.c:** Arquivo contendo a implementação do tratamento genérico dos comandos AT

**asf/libs/src/serial.c:** Arquivo contendo a implementação da porta serial utilizada pelas aplicações.

**asf/libs/src/serial2.c:** Arquivo contendo a implementação da porta serial auxiliar, utilizada para comunicação com os módulos.

**asf/libs/inc/at.h:** Arquivo de cabeçalho da implementação do tratamento genérico dos comandos AT.

**asf/libs/inc/serial.h:** Arquivo de cabeçalho d a implementação da porta serial utilizada pelas aplicações.

**asf/libs/inc/serial2.h:** Arquivo de cabeçalho da implementação da porta serial auxiliar, utilizada para comunicação com os módulos.

Foi criado ainda um projeto de teste para validação das 3 bibliotecas criadas. Esse projeto de testes foi evoluído e se transformou posteriormente na implementação da primeira aplicação de exemplo (tunnelTcp).

**PEDRO GRANVILLE**

Foi criada a biblioteca de comandos AT genérica para a plataforma Arduino já dentro da estrutura do projeto para o módulo HE910. Foi criado também um projeto de teste para a primeira implementação que evoluiu posteriormente para se transformar no exemplo tunnelTcp.

Os arquivos desenvolvidos foram:

**arduino/Telit\_GSM/AT.cpp:** Implementação dos médodos da biblioteca de comandos AT para arduino.

**arduino/Telit\_GSM/AT.h:** Definição da classe da biblioteca de comandos AT para arduino.

**ROOSEVELT VINÍCIUS**

Foi feito o levantamento e testes das sequências de comandos AT necessários para a conexão com a rede GSM/GPRS para o módulo HE910 e para conexão com a rede WIFI para o módulo GS2101M. Esses comandos serão utilizados em seguida para iniciar as implementações dos primeiros exemplos de aplicações.

**5. Atividade:** Implementação de Firmware Base

**Período de Execução:** 01/01/2018 a 31/01/2018

**Horas na Atividade:** 7

**Justificativa de participação na atividade:**

**MARIANA LINS**

A partir do projeto de testes implementado na etapa de desenvolvimento das bibliotecas de comandos AT, foi criado o projeto do primeiro exemplo de aplicação. Esse firmware deverá ser utilizado como base para o desenvolvimento dos projetos seguintes.

A partir da estrutura previamente definida, e das implementações de testes o FreeRTOS e comandos AT, foi criado o firmware. Entretanto ainda houveram diversos desafios, como a correta utilização e melhoria das implementações do FreeRTOS e ajustes e correções de bugs nas implementações de porta serial e comandos AT.

Por fim se chegou a uma versão satisfatória para termos como base para o desenvolvimento dos exemplos de aplicações.

**PEDRO GRANVILLE**

O firmware base foi desenvolvido a partir do firmware de testes utilizado no desenvolvimento da biblioteca de comandos AT. Poucos ajustes precisaram ser feitos já que a plataforma Arduino é simplista e de fácil implementação.

**ROOSEVELT VINÍCIUS**

Foi feito o levantamento dos comandos AT dos módulos HE910 e GS2101M para cada função:

* Inicialização do módulo
* Conexão na rede
* Conexão TCP
* Envio de dados TCP
* Recebimento de dados TCP

Após esse levantamento foram feitos testes para validação dos comandos utilizando a ferramenta de software Telit AT Controller.

**6. Atividade:** Testes dos Módulos e Montagem dos Ambientes de Desenvolvimento - Parte II

**Período de Execução:** 01/01/2018 a 31/01/2018

**Horas na Atividade:** 30

**Justificativa de participação na atividade:**

**MARIANA LINS**

Utilizando o firmware base feito para a plataforma Atmel Studio, foram feitos novos testes com os kits de desenvolvimento EVK2 e GainSpan GS2100M EVB3, e feito um firmware de testes implementando os comandos AT básicos de inicialização do dois módulos (HE910 e GS2101M).

Esse firmware de testes servirá como base para implementação das bibliotecas dos módulos.

**PEDRO GRANVILLE**

Com o firmware base feito para a plataforma Arduino, foram efetuados testes com os kits de desenvolvimento EVK2 e GainSpan GS2100M EVB3, e feito um firmware de testes implementando os comandos AT básicos de inicialização do dois módulos (HE910 e GS2101M).

Esse firmware de testes servirá como base para implementação das bibliotecas dos módulos.

**ROOSEVELT VINÍCIUS**

Foi dado o suporte para Mariana Lins e Pedro Granville para configuração de hardware, configuração e utilização dos kits de desenvolvimento EVK2 e GainSpan GS2100M EVB3.

A maior parte do esforço dedicado nesta atividade foi o envio correto e tratamento das respostas dos comandos AT de inicialização dos módulos e conexão com as redes GSM e Wifi.

**7. Atividade:** Implementação das bibliotecas dos módulos

**Período de Execução:** 01/02/2018 a 28/02/2018

**Horas na Atividade:** 8

**Justificativa de participação na atividade:**

**MARIANA LINS**

Criação inicial das bibliotecas de cada módulo, que implementam a *task* de gerenciamento de comunicação e funções que proveem as funcionalidades dos módulos para a camada de aplicação, utilizando a plataforma de software Atmel Studio (ASF+FreeRTOS).

Foram criados os seguintes códigos fontes:

**asf/libs/src/gs2101m.c:** Implementação das funções que disponibilizam as funcionalidades do módulo WIFI GS2101M.

**asf/libs/src/he910.c:** Implementação das funções que disponibilizam as funcionalidades do módulo GSM HE910.

**asf/libs/inc/gs2101m.h:** Arquivo de cabeçalho das funções que disponibilizam as funcionalidades do módulo WIFI GS2101M.

**asf/libs/inc/he910.h:** Arquivo de cabeçalho das funções que disponibilizam as funcionalidades do módulo GSM HE910.

**PEDRO GRANVILLE**

Foram implementadas as classes das bibliotecas para os módulos HE910 e GS2101M. Por enquanto foram feitas apenas os métodos de inicialização e conexão nas redes. O restante das funcionalidades serão implementadas de acordo com a necessidade, durante o desenvolvimento das aplicações de exemplo.

Foram criados os seguintes códigos fontes:

arduino/Telit\_GSM/HE910G.cpp

arduino/Telit\_GSM/HE910G.h

arduino/Telit\_Wifi/GS2101M.cpp

arduino/Telit\_Wifi/GS2101M.h

**ROOSEVELT VINÍCIUS**

Foi feito testes e validações em todos os comandos AT dos módulos HE910 e GS2101M utilizados nas implementações feitas por Mariana Lins e Pedro Granville. As sequência de comandos para cada função foram sendo validadas uma a uma antes de passar para eles.

**8. Atividade:** Implementação da Aplicação SMS Commands

**Período de Execução:** 01/02/2018 a 28/02/2018

**Horas na Atividade:** 8

Justificativa de participação na atividade:

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 lê continuamente as mensagens SMS recebidas pelo módulo, interpretando o seu conteúdo e executando comandos conforme o texto enviado pela mensagem, com resposta por SMS.

Foi criado o projeto de exemplo para a plataforma Atmel Studio chamado SMSCommands e uma task do FreeRTOS que implementa a aplicação.

Foram implementadas também as funções de envio e recebimento de SMS na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 lê continuamente as mensagens SMS recebidas pelo módulo, interpretando o seu conteúdo e executando comandos conforme o texto enviado pela mensagem, com resposta por SMS.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GSM\_HE910G\_SMSCommands.ino.

Foram implementadas também as funções de envio e recebimento de SMS na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 lê continuamente as mensagens SMS recebidas pelo módulo, interpretando o seu conteúdo e executando comandos conforme o texto enviado pela mensagem, com resposta por SMS.

Foi feito testes, validações e acompanhamento da implementação dos comandos de envio e recebimento de SMS para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**9. Atividade:** Implementação da Aplicação Túnel TCP

**Período de Execução:** 01/02/2018 a 28/02/2018

**Horas na Atividade:** 39

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu (contexto com a APN ou o socket) e tenta continuamente uma reconexão. Permite também o fluxo de dados servidor->dispositivo, refletido na porta serial. Um túnel de porta serial por TCP.

No desenvolvimento dessa aplicação, foram consolidados a utilização efetiva do FreeRTOS e das implementações de acesso às portas seriais na plataforma Atmel Studio. A aplicação foi desenvolvida para ser o modelo para as próximas.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado tunnelTcp e a task FreeRTOS contendo a lógica da aplicação do túnel, chamado tunnelTcp.c.

Foram implementadas também as funções de conexão, escrita, leitura e desconexão TCP na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu (contexto com a APN ou o socket) e tenta continuamente uma reconexão. Permite também o fluxo de dados servido<-->dispositivo, refletido na porta serial. Um tunel de porta serial por TCP.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GSM\_HE910G\_UartSocketTunnel.ino.

Foram implementadas também as funções de conexão, escrita, leitura e desconexão TCP na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu (contexto com a APN ou o socket) e tenta continuamente uma reconexão. Permite também o fluxo de dados servidor->dispositivo, refletido na porta serial. Um tunel de porta serial por TCP.

Foram feitos testes, validações e acompanhamento da implementação dos comandos de abertura de conexão, escrita, leitura e desconexão do protocolo TCP para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**10. Atividade:** Implementação da Aplicação Porteiro Eletrônico

**Período de Execução:**01/02/2018 a 28/02/2018

**Horas na Atividade:**20

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 implementa um sistema de porteiro eletrônico que, se um botão for pressionado, liga para um número de telefone predefinido e estabelece uma comunicação de voz entre eles.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado intercomGsm e a task FreeRTOS contendo a lógica da aplicação do túnel, chamado intercomGsm.c.

Foram implementadas também as funções de chamadas de voz e atendimento de chamadas na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 implementa um sistema de porteiro eletrônico que, se um botão for pressionado, liga para um número de telefone predefinido e estabelece uma comunicação de voz entre eles.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GSM\_HE910G\_Porteiro\_Eletronico.ino.

Foram implementadas também as funções de chamadas de voz e atendimento de chamadas na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 implementa um sistema de porteiro eletrônico que, se um botão for pressionado, liga para um número de telefone predefinido e estabelece uma comunicação de voz entre eles.

Foi feito testes, validações e acompanhamento da implementação dos comandos de chamada de voz para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**11. Atividade:** Implementação da Aplicação LBS Geofence SMS

**Período de Execução:** 01/03/2018 a 31/03/2018

**Horas na Atividade:** 8

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 implementa uma associação entre o sistema de geofence com envio de posições lidas a partir do LBS através do sistema de comunicação por SMS.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado lbsGeofenceSms e a task FreeRTOS contendo a lógica da aplicação, chamado lbsGeofenceSms.c.

Foram implementadas também as funções de localização através de LBS na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 implementa uma associação entre o sistema de geofence com envio de posições lidas a partir do LBS através do sistema de comunicação por SMS.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GPS\_HE910G\_LBS\_Geofence\_SMS.ino.

Foram implementadas também as funções de localização através de LBS na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 implementa uma associação entre o sistema de geofence com envio de posições lidas a partir do LBS através do sistema de comunicação por SMS.

Foi feito testes, validações e acompanhamento da implementação dos comandos de localização através de LBS para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**12. Atividade:** Implementação da Aplicação GPS Geofence

**Período de Execução:**01/03/2018 a 31/03/2018

**Horas na Atividade:**14

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 determina o pertencimento de uma latitude/longitude recebida através dos códigos NMEA emitidos automaticamente pelo módulo GPS, pela porta serial, com o interior polígono definido através de um array de coordenadas. Quando o fix, representado pela mensagem $GPGGA, está fora do polígono, envia onda quadrada para um buzzer/falante e acende um LED.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado gpsGeofence e a task FreeRTOS contendo a lógica da aplicação, chamado gpsGeofence.c.

Foram implementadas também as funções de localização através da tecnologia GPS na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 determina o pertencimento de uma latitude/longitude recebida através dos códigos NMEA emitidos automaticamente pelo módulo GPS, pela porta serial, com o interior polígono definido através de um array de coordenadas. Quando o fix, representado pela mensagem $GPGGA, está fora do polígono, envia onda quadrada para um buzzer/falante e acende um LED.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GPS\_HE910G\_GPS\_Geofence.ino.

Foram implementadas também as funções de localização através da tecnologia GPS na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 determina o pertencimento de uma latitude/longitude recebida através dos códigos NMEA emitidos automaticamente pelo módulo GPS, pela porta serial, com o interior polígono definido através de um array de coordenadas. Quando o fix, representado pela mensagem $GPGGA, está fora do polígono, envia onda quadrada para um buzzer/falante e acende um LED.

Foi feito testes, validações e acompanhamento da implementação dos comandos de localização através da tecnologia GPS para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**13. Atividade:** Implementação da Aplicação MQTT LBS

**Período de Execução:**01/03/2018 a 31/03/2018

**Horas na Atividade:**11

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 é uma associação do sistema de geofence com publicação de posições lidas a partir do LBS em um servidor MQTT.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado lbsMqtt e a task FreeRTOS contendo a lógica da aplicação, chamado lbsMqtt.c.

Foram implementadas também as funções de conexão, escrita, leitura e desconexão MQTT na biblioteca he910.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 é uma associação do sistema de geofence com publicação de posições lidas a partir do LBS em um servidor MQTT.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GSM\_UE910GL\_HARDWARE\_MQTT\_LBS.ino.

Foram implementadas também as funções de conexão, escrita, leitura e desconexão MQTT na biblioteca HE910.cpp.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 é uma associação do sistema de geofence com publicação de posições lidas a partir do LBS em um servidor MQTT.

Foi feito testes, validações e acompanhamento da implementação dos comandos de conexão, leitura, escrita e desconexão MQTT para o módulo HE910. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**14. Atividade:** Implementação da Aplicação GPS Geofence SMS

**Período de Execução:**01/03/2018 a 31/03/2018

**Horas na Atividade:**9

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo HE910 é uma associação entre o sistema de geofence com envio de posições lidas a partir do GPS através do sistema de comunicação por SMS.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado gpsGeofenceSms e a task FreeRTOS contendo a lógica da aplicação, chamado gpsGeofenceSms.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo HE910 é uma associação entre o sistema de geofence com envio de posições lidas a partir do GPS através do sistema de comunicação por SMS.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GPS\_HE910G\_GPS\_Geofence\_SMS.ino.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo HE910 é uma associação entre o sistema de geofence com envio de posições lidas a partir do GPS através do sistema de comunicação por SMS.

Foi feita a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**15. Atividade:** Implementação da Aplicação MQTT

**Período de Execução:**01/03/2018 a 31/03/2018

**Horas na Atividade:**14

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo GS2101M publica variáveis em sistema MQTT que lê uma fila de dados a serem enviados para o portal MQTT da Telit, em pares "nome da variável" e "valor", e para cada item executa os comandos de post de variáveis por MQTT.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado mqtt e a task FreeRTOS contendo a lógica da aplicação, chamado mqtt.c.

Foi feita a integração da biblioteca Paho Embedeed MQTT C/C++ (http://www.eclipse.org/paho/clients/c/embedded/) com o projeto, para a implementação de protocolo MQTT por software.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo GS2101M publica variáveis em sistema MQTT que lê uma fila de dados a serem enviados para o portal MQTT da Telit, em pares "nome da variável" e "valor", e para cada item executa os comandos de post de variáveis por MQTT.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_GSM\_GS2101M\_MQTT.ino.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo GS2101M publica variáveis em sistema MQTT que lê uma fila de dados a serem enviados para o portal MQTT da Telit, em pares "nome da variável" e "valor", e para cada item executa os comandos de post de variáveis por MQTT.

Foi feita a especificação do funcionamento da aplicação, testes e documentação no Wiki.

**16. Atividade:** Implementação da Aplicação Túnel TCP

**Período de Execução:**01/03/2018 a 31/03/2018

**Horas na Atividade:**19

**Justificativa de participação na atividade:**

**MARIANA LINS**

Essa aplicação de exemplo para o módulo GS2101M é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu e tenta continuamente uma reconexão. Permite também o fluxo de dados servidor->dispositivo, refletido na porta serial. Um túnel de porta serial por TCP.

Foram criados o projeto Atmel Studio da aplicação de exemplo chamado tunnelTcp e a task FreeRTOS contendo a lógica da aplicação do túnel, chamado tunnelTcp.c.

Foram implementadas também as funções de conexão, escrita, leitura e desconexão TCP na biblioteca gs2101m.c.

**PEDRO GRANVILLE**

Essa aplicação de exemplo para o módulo GS2101M é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu e tenta continuamente uma reconexão. Permite também o fluxo de dados servidor->dispositivo, refletido na porta serial. Um túnel de porta serial por TCP.

Foi criado o exemplo para a plataforma Arduino que implementa a aplicação chamado Telit\_Wifi\_GS2101M\_UartTcpTunnel.ino.

**ROOSEVELT VINÍCIUS**

Essa aplicação de exemplo para o módulo GS2101M é um sistema de logger baseado em modem que repassa o tráfego de uma das portas seriais da placa Arduino para um socket aberto com um servidor remoto, através dos comandos AT para sockets. Bufferiza os bytes que são recebidos enquanto a conexão não estava ativa. As configurações de servidor, baudrate e afins são manuais, através de constantes no código. O sistema verifica que a conexão caiu e tenta continuamente uma reconexão. Permite também o fluxo de dados servidor->dispositivo, refletido na porta serial. Um túnel de porta serial por TCP.

Foi feito testes, validações e acompanhamento da implementação dos comandos de conexão, leitura, escrita e desconexão TCP para o módulo GS2101M. Foi feito ainda a especificação do funcionamento da aplicação, testes e documentação no Wiki.