

Smart mail box®

Relatório

Documento elaborado por:
Cláudio Esperança – aluno n.º 2120917
Diogo Serra – aluno n.º 2120915

Docentes:
Nuno Costa

Versão 1.0.71
10 de junho de 2013

Revisões

Versão	Autor(es)	Descrição das alterações
1.0	Cláudio Esperança Diogo Serra	<ul style="list-style-type: none">• Versão inicial do documento

Índice de conteúdos

1. Abstract.....	1
2. Introdução.....	2
3. Estado da arte.....	3
4. Especificações.....	5
4.1. Requisitos.....	5
4.1.1. Terminologia.....	5
4.1.2. Requisitos funcionais.....	6
4.1.3. Requisitos não funcionais.....	6
4.1.4. Requisitos de desenvolvimento.....	6
4.2. Arquitetura.....	6
4.2.1. Hardware.....	7
4.2.2. Software.....	8
5. Implementação do protótipo.....	16
5.1. Hardware.....	16
5.2. Software.....	22
5.2.1. Módulo 1 – nó/caixa de correio.....	23
5.2.2. Módulo 2 – sistema central.....	23
6. Avaliação e análise.....	25
7. Conclusões e trabalhos futuros.....	27

Índice de tabelas

Tabela 1: Terminologia.....	5
Tabela 2: Requisitos funcionais.....	6
Tabela 3: Requisitos não funcionais.....	6
Tabela 4: Lista de hardware proposto.....	8
Tabela 5: Lista de hardware do protótipo.....	16

Índice de imagens

Imagen 1: Arquitetura da solução.....	7
Imagen 2: Algoritmo A1 - configuração do nó.....	9
Imagen 3: Algoritmo A2 - notificação de correio.....	11
Imagen 4: Algoritmo A3 - reposição e envio manual de notificação de estado.....	12
Imagen 5: Algoritmo A4 - notificação automática e periódica de estado.....	14
Imagen 6: Fotografia do protótipo do sistema central.....	17
Imagen 7: Esquema de ligações utilizado entre os componentes.....	18
Imagen 8: Esquema de ligações concretizado na breadboard.....	19
Imagen 9: Instalação do equipamento na "caixa de correio".....	20
Imagen 10: Aspetto final da "caixa do correio".....	21
Imagen 11: Alguns dos comandos disponíveis no módulo do sistema central.....	24
Imagen 12: Registo da informação obtida num dos testes efetuados.....	25

Lista de acrónimos

Para referência deixamos uma lista de acrónimos utilizados ao longo do documento:

Acrónimo	Designação
ACK	<i>Acknowledge</i>
API	<i>Application Programming Interface</i>
DEI	Departamento de Engenharia Informática
EEPROM	<i>Electrically Erasable Programmable Read Only Memory</i>
ESTG	Escola Superior de Tecnologia e Gestão de Leiria
GPL	<i>GNU General Public License</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IPL	Instituto Politécnico de Leiria
JS	<i>JavaScript</i>
MEI	Mestrado em Engenharia Informática
MEI-CM	Mestrado em Engenharia Informática – Computação Móvel
PHP	<i>PHP: Hypertext Preprocessor</i>
PIR	<i>Passive InfraRed sensor</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
TLS	<i>Transport Layer Security</i>
UC	Unidade Curricular
UI	<i>User Interface</i>

1. Abstract

In a world that moves increasingly at the speed of the Internet and where time is becoming a more and more valuable resource, the introduction of smart objects in everyday life can save time and increase the quality of people's lives. As engineers for the future, it is important to look around us and see how we can turn the most common object into a smart object, increasing and potentiating their value. In this project we will present our proposal for building a smart mailbox able to detect the introduction of correspondence and notify a remote control system of the state change.

2. Introdução

Este documento apresenta o trabalho desenvolvido no âmbito do projeto da unidade curricular de Mobilidade em Redes de Comunicação do Mestrado em Engenharia Informática - Computação Móvel, da Escola Superior de Tecnologia e Gestão para o ano letivo de 2012/2013.

Estima-se que em 2012 o volume de itens de correio processados a nível mundial tenha sido na ordem dos 400 mil milhões [1]. Em plena era digital, com a disseminação das comunicações eletrónicas, o volume de correspondência tradicional tem vindo a diminuir. No entanto, com a Internet apareceram novas oportunidades introduzidas pelo comércio eletrónico que dependem dos serviços postais para fazer a entrega das encomendas que fazemos nas lojas digitais.

O principal objetivo deste projeto consiste no desenvolvimento de uma *smart mail box*, um objeto inteligente que fornece informações adicionais sobre o seu estado. Através destas informações um utilizador poderá saber se tem correspondência na sua caixa de correio física.

Este documento apresenta a solução proposta bem como as diferentes fases no desenvolvimento de um protótipo funcional da *smart mail box*.

No capítulo 3 falaremos um pouco sobre o estado da arte e no capítulo 4 abordaremos as especificações da solução proposta, tais como os requisitos e a arquitetura da solução. O protótipo implementado e os testes realizados serão abordados, respetivamente nos capítulos 5 e 6. As habituais conclusões serão apresentadas no capítulo 7.

3. Estado da arte

Os *smarts objects* (objetos inteligentes) surgem no contexto da computação ubíqua onde o processamento da informação passa a estar integrada em objetos comuns [2], que fornecem informação adicional sobre o seu estado [3], perfeitamente integrados no meio humano, simplificando e/ou potenciando atividades.

Neste contexto surge a *smart mail box* como uma caixa de correio inteligente capaz de detetar a introdução de correio físico e notificar o ambiente dessa alteração de estado. O conceito de *smart mail box* não é novo e o seu funcionamento geral pode ser descrito na patente US 7256691 B2 atribuída a Christopher Awobue [4], com relação às patentes para indicação remota da entrega de correio (US 5950919 A - [5]) e sistema para deteção de correio numa caixa de correio (US 20050122220 A1 - [6]).

Existem no mercado alguns produtos deste tipo como a *STI Wireless MailBox Alert* da empresa *Safety Technology International*, um produto destinado essencialmente ao mercado norte americano e que pode ser instalado em caixas de correio exteriores existentes. Utiliza um sensor magnético para detetar que a porta da caixa de correio foi aberta e assim sinalizar a existência de correio.

Mike Riley [7] propõe outra solução no seu *Package Delivery Detector* onde, através de um sensor de pressão, um arduino envia uma notificação quando uma encomenda é colocada sob o sensor.

Ambas as soluções apresentam algumas limitações. No caso do *STI Wireless MailBox Alert* [8], o sensor apenas deteta que o compartimento de acesso à caixa de correio foi aberto o que obriga à existência um sistema mecânico para a ativação do sistema, limitando a sua aplicação e permitindo a existência de falsos positivos (alguém que abriu a caixa de correio mas não colocou correspondência).

O *Package Delivery Detector* está limitado a encomendas que precisam de ter um peso mínimo (mais de 500g) para ativar o sistema, excluindo assim a correspondência mais comum.

Assim faz sentido a implementação de uma *smart mail box* que não sofra destas limitações permitindo a deteção de correio de forma transparente e eficaz.

4. Especificações

4.1. Requisitos

Neste projeto os principais objetivos são o planeamento, desenho e implementação de uma solução para uma caixa de correio inteligente que notifique um sistema central para a receção de correspondência. Foram definidos os seguintes requisitos:

4.1.1. Terminologia

Termo	Descrição
<i>Smart object / objeto inteligente</i>	Objeto com capacidade de processamento que fornece informação sobre o seu estado
<i>Caixa de correio</i>	Caixa que irá receber o hardware necessário para que possa ser transformada num objeto inteligente
<i>Smart mail box / caixa de correio inteligente</i>	Caixa de correio que foi convertida num objeto inteligente
<i>Utilizador</i>	Utilizador que interage com a solução
<i>Sistema central</i>	Serviço que recebe e processa as notificações de eventos com origem na caixa de correio inteligente
<i>Eventos</i>	Operação de retirada ou de introdução de correio

Tabela 1: Terminologia

4.1.2. Requisitos funcionais

Referência	Descrição	Prioridade
RF001	O sistema deve detetar a introdução de <u>correo postal</u> na <i>caixa de correio</i>	Alta
RF002	O sistema deve detetar a remoção de <u>correo postal</u> na <i>caixa de correio</i>	Baixa
RF003	A <i>caixa de correio inteligente</i> deve notificar o <i>sistema central</i> de <u>eventos</u> .	Alta

Tabela 2: Requisitos funcionais

4.1.3. Requisitos não funcionais

Referência	Descrição	Prioridade
RNF001	O hardware a instalar na caixa de correio deve ser compacto	Média
RNF002	A alimentação do hardware a instalar na caixa deve ser autónoma e não deve depender da rede elétrica existente	Média
RNF003	A transmissão de informação entre a caixa de correio inteligente e o sistema central deve ser baseada em comunicações sem fios	Alta

Tabela 3: Requisitos não funcionais

4.1.4. Requisitos de desenvolvimento

Para este sistema não foram propostos requisitos de desenvolvimento.

4.2. Arquitetura

A proposta de arquitetura para a solução encontra-se representada na imagem 1.

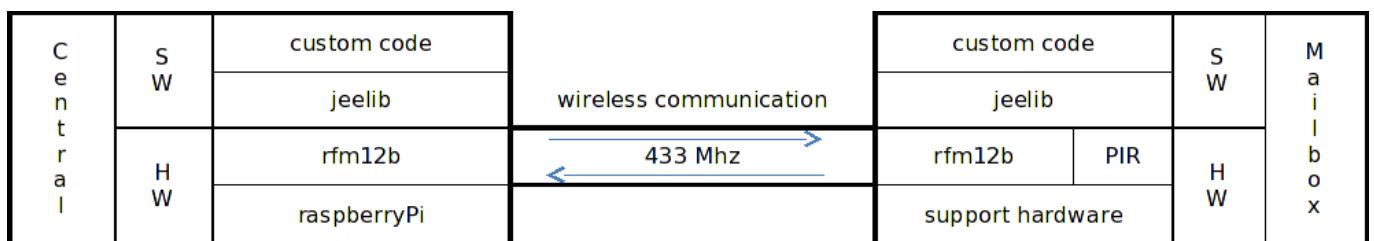


Imagen 1: Arquitetura da solução

4.2.1. Hardware

Do ponto de vista do *hardware*, a caixa de correio (nó) deve ser composta por um sensor de movimento (PIR), um transmissor/recetor sem fios (rfm12b) e pelo restante *hardware* de suporte que forneça a infraestrutura necessária para a execução das instruções que definem o comportamento da solução. No sistema central deverá também existir um recetor sem fios (rfm12b) que fará a receção e interpretação das mensagens enviadas pela caixa de correio. Este módulo pode estar acoplado a um *Raspberry Pi* ou outro sistema que possa processar as mensagens recebidas e executar ações em conformidade com as mesmas.

Caixa de correio	Central
<ul style="list-style-type: none"> • 1x HAHnode PCB • 1x AVR microcontroller • 1x AVR socket • 1x RFM12B (433/868 Mhz) • 1x resonator • 2x resistor • 1x voltage regulator • 1x electrolytic capacitor • 4x ceramic capacitors • 1x 1/4 wave antenna wire • 1x LDR • 1x white rubber grommet • 1x 2way JST PCB connector + 	<ul style="list-style-type: none"> • 1x ATmega328 (running OpiBoot serial bootloader) • 2x 100nF • 1x 5mm LED • 2x 10K (brown, black, orange, gold) • 1x 100R (brown, black, brown, gold) • 1x RFM12B (433/868 Mhz) • 1x GPIO 6x2 pin header female socket • 1x ISP 3x2 pin header • 1x Rpi

Caixa de correio	Central
<ul style="list-style-type: none"> matching cable • 2x 3way JST PCB connector + matching cable • 1x length of clear heatshrink • 1x Pir Sensor 	

Tabela 4: Lista de *hardware* proposto

Uma pesquisa de mercado por alguns fornecedores para a lista de *hardware* mencionada na tabela 4 permitiu situar os custos dos componentes em aproximadamente 69,3€ (23,1€ para o *hardware* da caixa e 46,2€ para o sistema central – incluindo o *Raspberry PI*). Apesar de este valor não incluir margens de lucro nem os custos de produção é expetável que, com a produção em massa, este custo com os componentes desça consideravelmente.

A inclusão de um *Raspberry PI* aumenta consideravelmente o preço da solução. No entanto este componente é opcional e, em sua substituição, pode ser integrado qualquer outro componente com um módulo RFM12b que faça o processamento dos dados enviados pelo nó na caixa de correio. No entanto a inclusão deste equipamento não foi inocente, dado que este componente permitirá aumentar muito facilmente a escabilidade do sistema, podendo ser integrados outros sensores para desempenho de outras funções auxiliares numa casa inteligente.

4.2.2. Software

Relativamente ao *software* sugere-se uma camada de abstração para a comunicação baseada na biblioteca *JeeLib*¹ responsável por controlar o módulo RF12 sem fios, temporizadores, gestão de energia e várias classes de ligação. O código personalizado, recorrendo à biblioteca *JeeLib* e ao *hardware* de suporte, será responsável pela implementação das funcionalidades descritas nos algoritmos propostos no âmbito desta solução.

¹ <https://github.com/jcw/jeelib>

Algoritmo A1: Configuração do nó

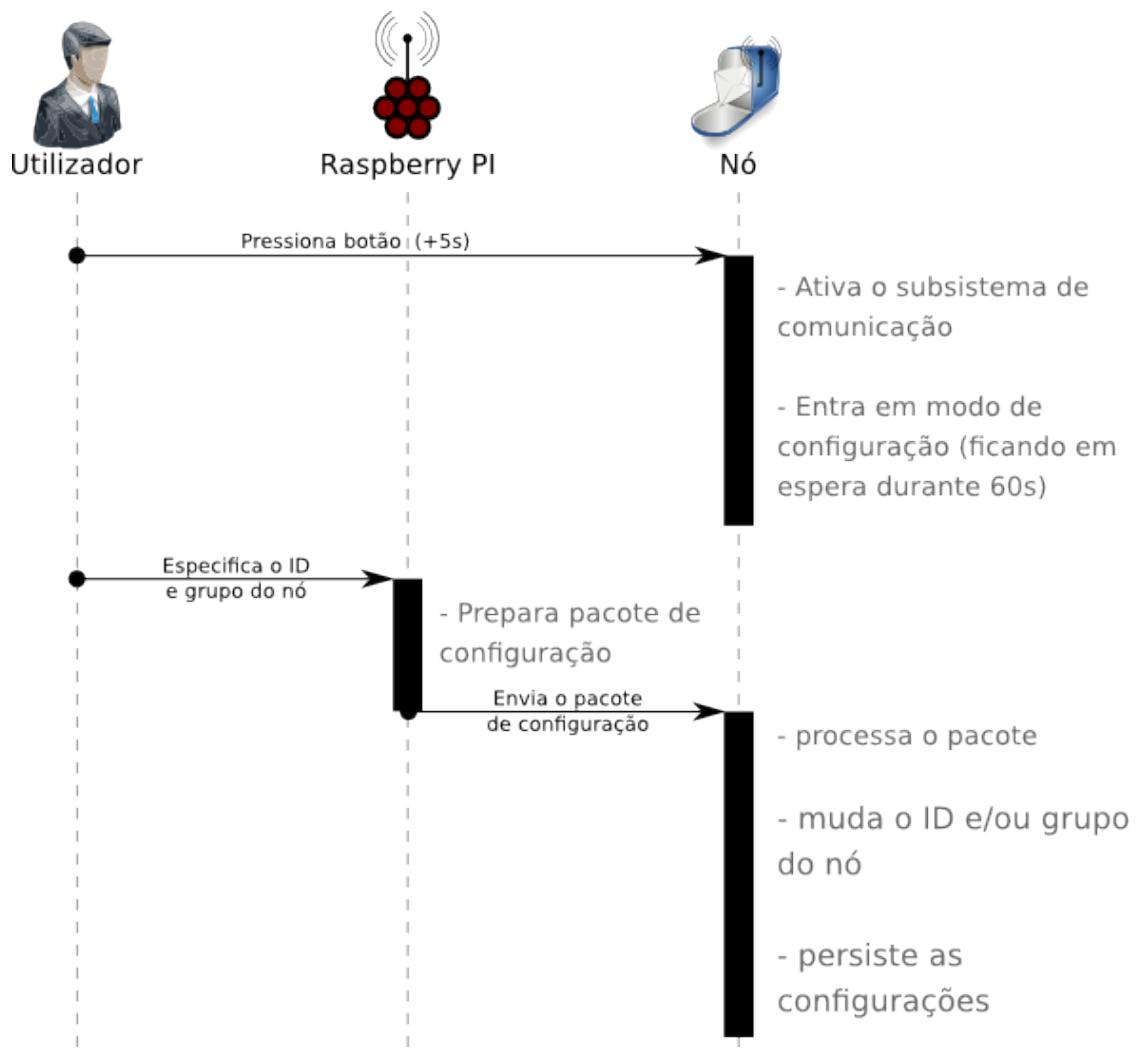


Imagen 2: Algoritmo A1 - configuração do nó

Por omissão o nó da caixa de correio está configurado com um identificador e grupo por omissão. Apesar de ser totalmente funcional, num ambiente concorrente onde poderão existir múltiplas caixas de correio com um equipamento semelhante, poderiam existir falsos positivos em que a notificação da presença de correio numa outra caixa seria reportada para outros sistemas não relacionados. Assim optou-se por definir um sistema que permita a configuração do identificador e grupo que identifica cada uma das caixas de correio.

Neste algoritmo, o utilizador começa por pressionar durante pelo menos 5 segundos o botão de configuração no equipamento da caixa de correio (nó). Este

equipamento ativa o subsistema de comunicação e o modo de configuração aguarda por um pacote de configuração durante 60 segundos. Caso não seja enviado nenhum pacote de configuração, são assumidas as configurações anteriores.

Numa segunda fase, e enquanto o nó está em modo de configuração, o utilizador introduz o identificador e grupo de origem bem como as novas definições para configuração do identificador e grupo do nó. Após a receção das configurações o sistema central (*Raspberry PI*) prepara o pacote de configuração com os dados fornecidos e envia o mesmo para o nó especificado.

Após a receção do pacote durante o intervalo de tempo de configuração, o nó processa os conteúdos do mesmo extraiendo a informação sobre o novo identificador e/ou grupo, assumindo e persistindo estas definições.

Algoritmo A2: Notificação de correio

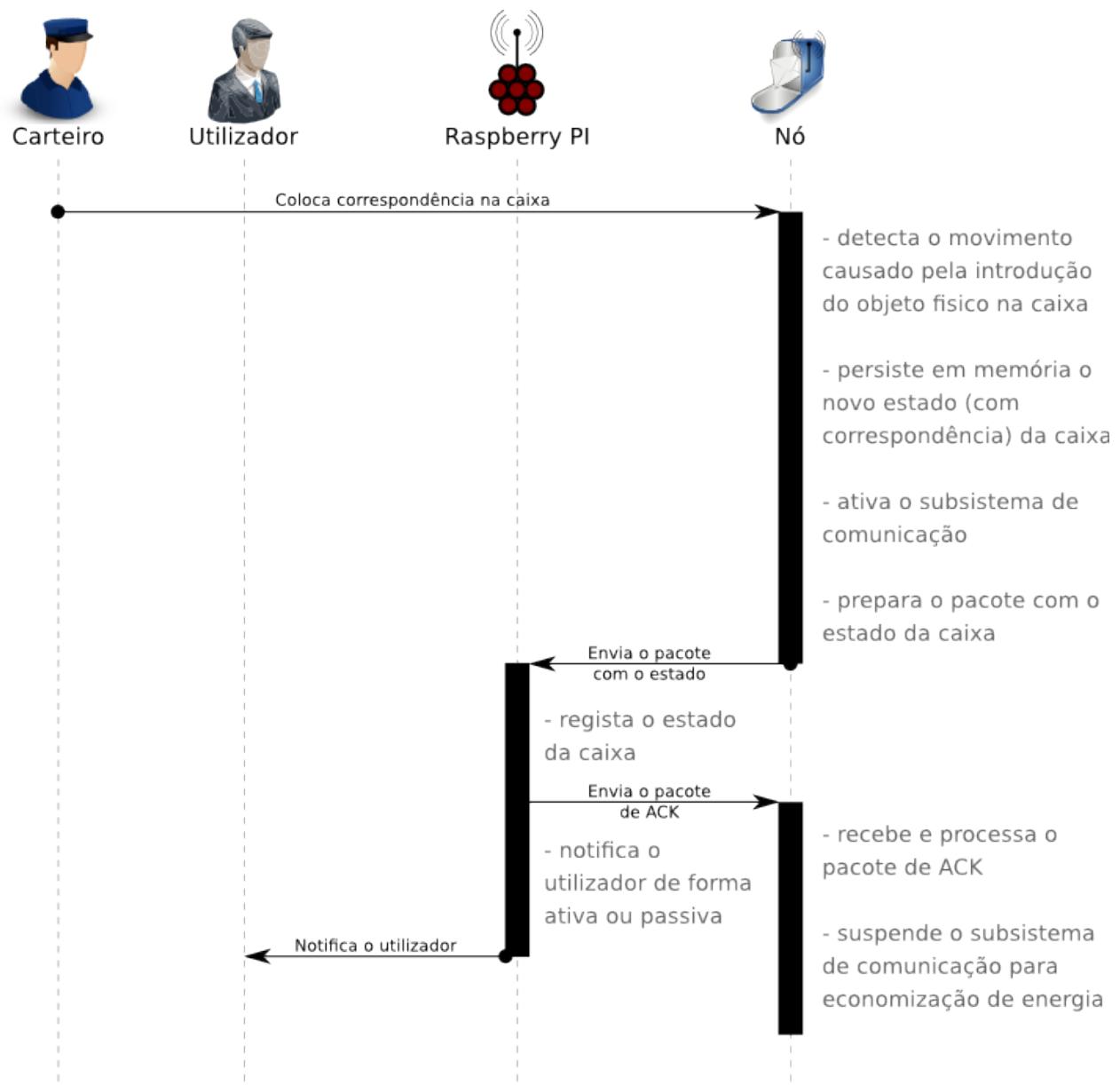


Imagen 3: Algoritmo A2 - notificação de correio

Neste algoritmo o carteiro coloca a correspondência na caixa de correio. O hardware do nó deteta o movimento causado pela introdução do objeto físico na caixa, persistindo em memória o novo estado da caixa (com correspondência). Após o registo desta nova informação, o sistema ativa o subsistema de comunicação, prepara o pacote com o novo estado da caixa e envia esta informação para o sistema central (*Raspberry PI*).

O sistema central regista o novo estado e envia o pacote de ACK para o nó que o

processa e suspende o subsistema de comunicação. Após o envio do pacote de ACK, o sistema central notifica o utilizador de forma ativa ou passiva de acordo com as definições do serviço.

Algoritmo A3: Reposição e envio manual de notificação de estado

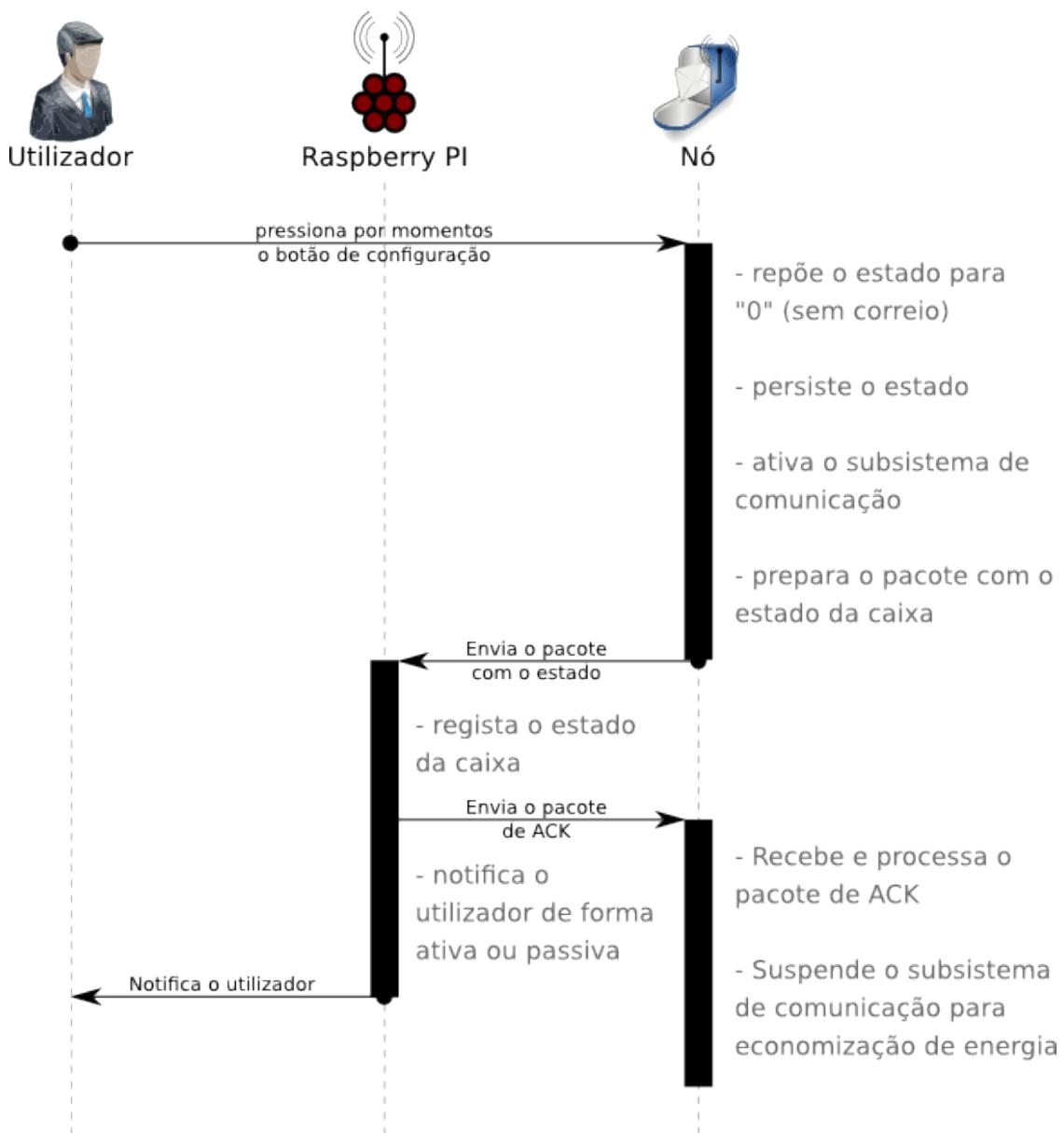


Imagen 4: Algoritmo A3 - reposição e envio manual de notificação de estado

Quando o correio é retirado da caixa é necessário informar o sistema central desta alteração de estado. Por outro lado, poderão existir situações onde é necessário confirmar se o sistema está operacional através do envio de notificação de estado

de modo manual. Este algoritmo fornece uma resposta a estes dois cenários através da utilização de uma única ação. Para o efeito, o utilizador pressiona por momentos (menos de 5 segundos) o botão de configuração no nó; o sistema altera o estado interno da caixa para sem correio, persistindo este estado em memória. Depois, o nó ativa o subsistema de comunicação, preparando e enviando um pacote com o estado da caixa e da bateria para o sistema central.

O sistema central recebe e processa o pacote registando o estado da caixa. De seguida envia o pacote de ACK e notifica o utilizador de forma ativa ou passiva de acordo com as definições do serviço. Por fim, o nó recebe o pacote de ACK enviado anteriormente e suspende o subsistema de comunicação para economização de energia.

Algoritmo A4: Notificação automática e periódica de estado

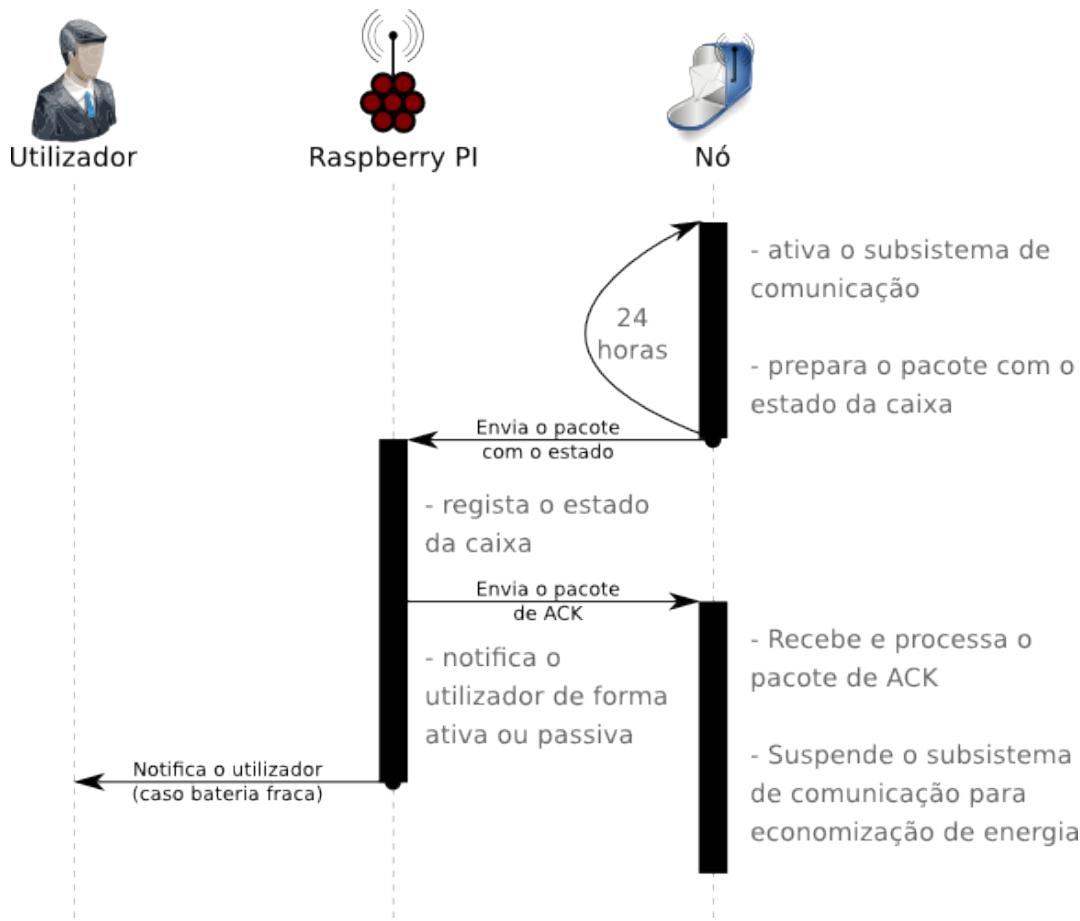


Imagen 5: Algoritmo A4 - notificação automática e periódica de estado

De acordo com os requisitos definidos para a solução, a alimentação do nó deve ser independente e autónoma (por exemplo, com recurso a pilhas). Caso a fonte de energia não seja renovável é recomendável a implementação de um mecanismo de notificação que informe o utilizador quando um nó não está ativo (por exemplo, quando as pilhas não conseguirem alimentar o equipamento). Neste sentido este algoritmo prevê que um nó deve notificar diariamente um sistema central sobre o seu estado para que, caso não seja recebida nenhuma mensagem do nó durante um período de 24 horas, sejam executados os procedimentos considerados necessários (por exemplo, notificar o utilizador da anomalia).

Assim, após ciclos de 24 horas, o nó ativa o subsistema de comunicação enviando o pacote com o estado atual da caixa para o sistema central.

Este regista o estado da caixa, notificando o utilizador de anomalias reportadas caso seja necessário. O sistema central envia ainda o pacote de ACK para o nó que o processa desativando de seguida o subsistema de comunicações para economização de energia.

5. Implementação do protótipo

Para validar a solução proposta foi construído um protótipo funcional onde foram implementados os algoritmos descritos.

5.1. Hardware

Para a implementação do protótipo foram utilizados os seguintes componentes:

Caixa de correio	Sistema central
<ul style="list-style-type: none">• LED vermelho – 3mm• Arduino Nano v2.3• Sensor PIR• Resistência 10k Ω• Resistência 220 Ω• Transcetor RFM12b HopeRF• Botão• <i>Breadboard</i>	<ul style="list-style-type: none">• <i>Raspberry PI</i>• <i>RFM2Pi board</i>• Cartão de memória

Tabela 5: Lista de hardware do protótipo

No *Raspberry PI* do sistema central foi instalada a placa RFM2Pi e o cartão de memória onde foi instalado e configurado o sistema operativo GNU/Linux que serviu de suporte à solução.

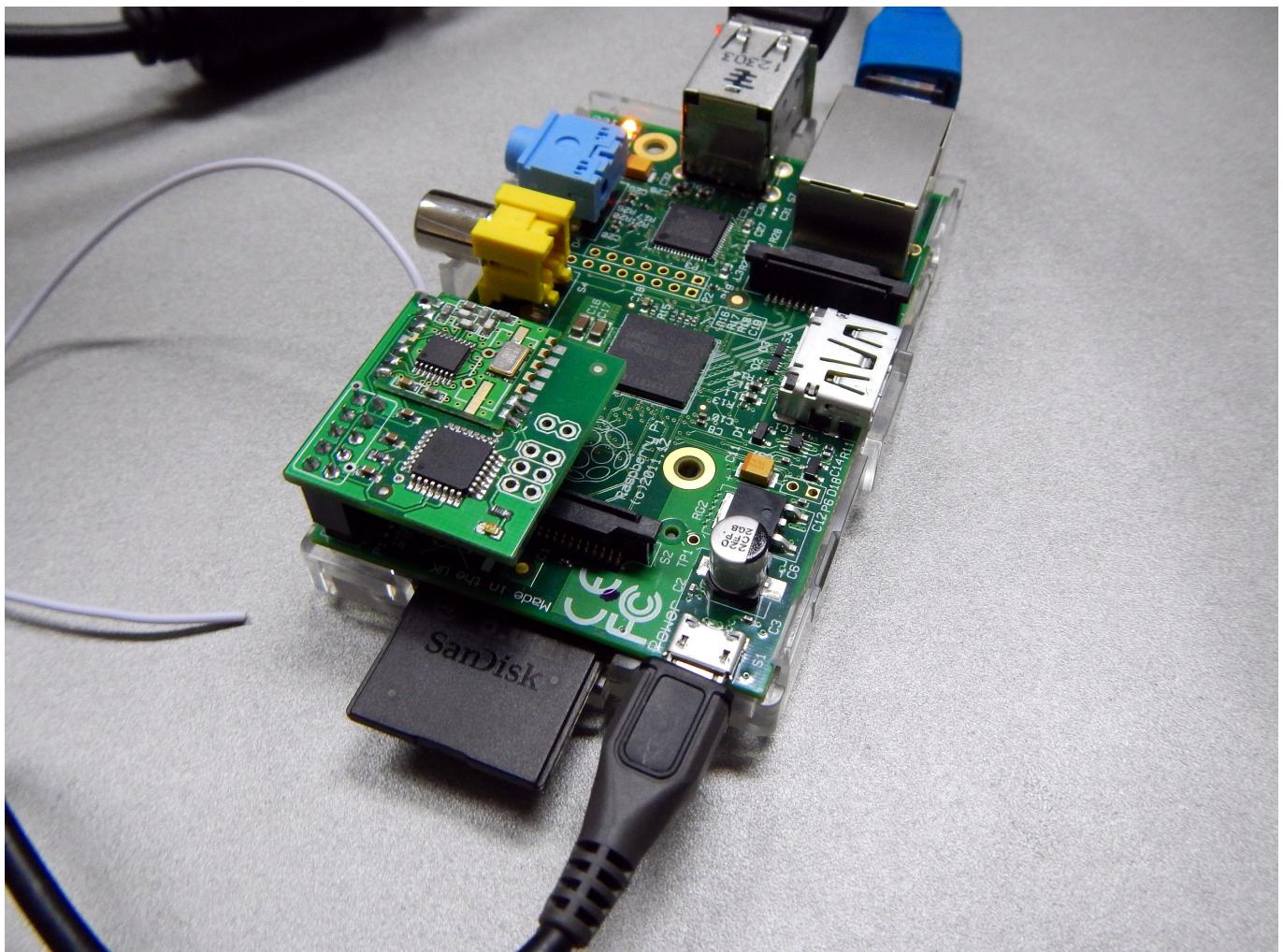


Imagen 6: Fotografia do protótipo do sistema central

O esquema de componentes da caixa de correio é apresentado na imagem 7.

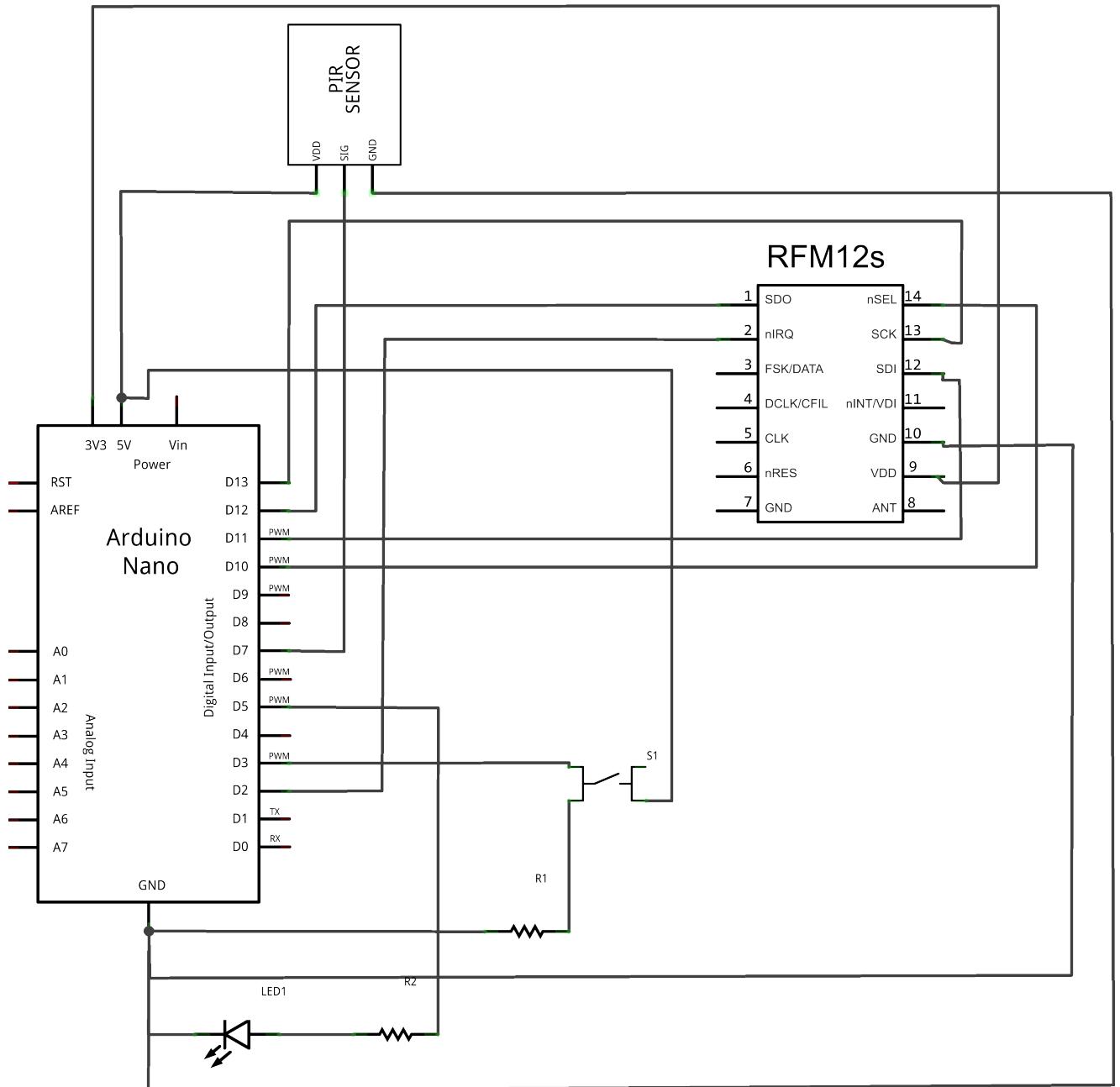


Imagen 7: Esquema de ligações utilizado entre os componentes

À esquerda temos o *arduino nano* como o controlador do nó, no topo o sensor de infravermelhos responsável pela deteção de movimentos e à direita o transceptor sem fios responsável pela comunicação. Ao centro temos o interruptor/botão de configuração, no final do esquema um LED de sinalização e as resistências necessárias para a proteção e funcionamento do circuito.

O esquema de ligações da imagem anterior foi posteriormente implementado sob uma *breadboard* num resultado semelhante ao da imagem 8.

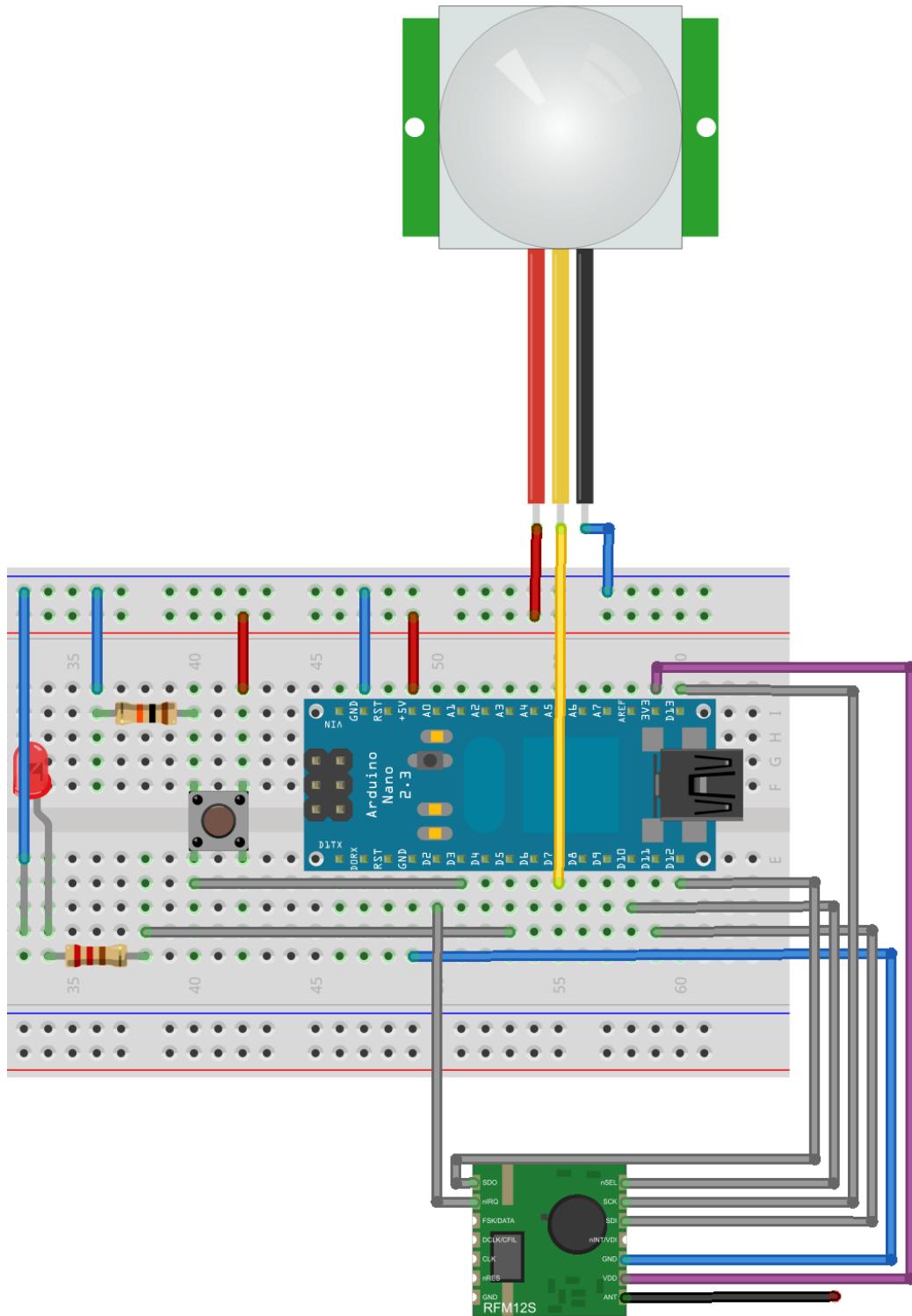


Imagen 8: Esquema de ligações concretizado na *breadboard*

Após todas as ligações efetuadas o *hardware* foi instalado numa caixa de cartão para simular as condições de uma caixa de correio.

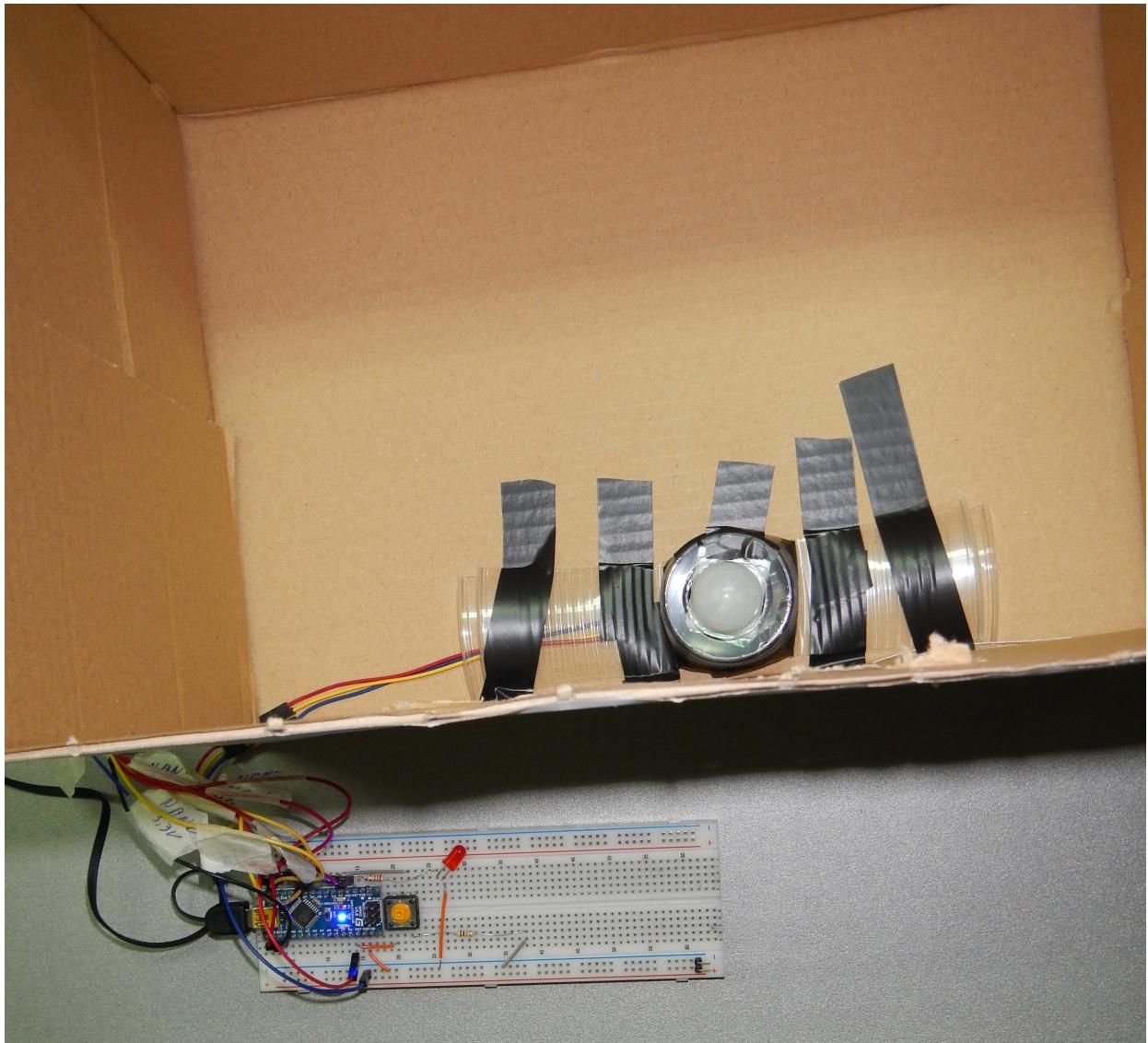


Imagen 9: Instalação do equipamento na "caixa de correio"

O aspeto final do protótipo da caixa é apresentado na imagem 10.

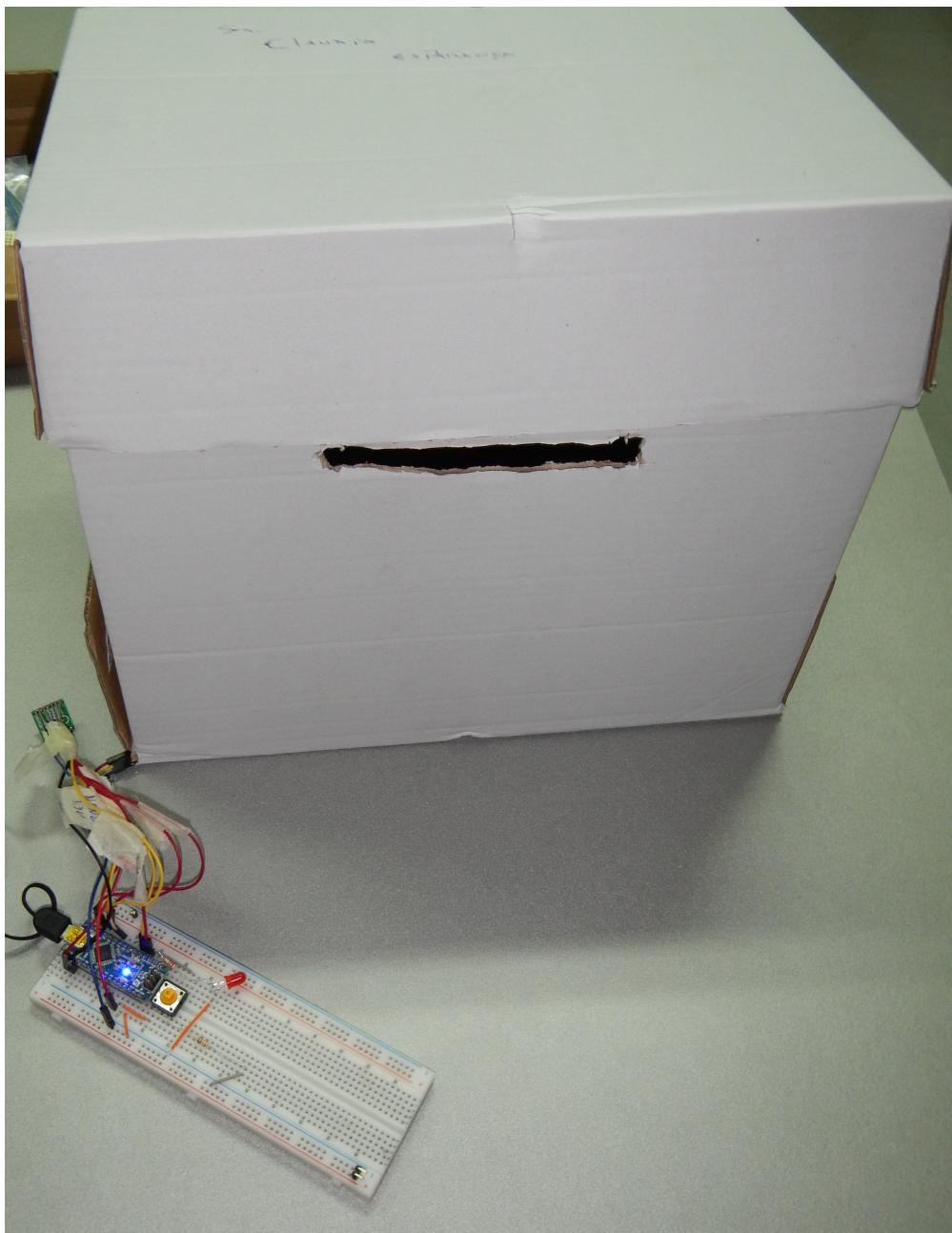


Imagen 10: Aspetto final da "caixa do correio"

5.2. Software

Para o desenvolvimento deste projeto optou-se por uma metodologia de desenvolvimento ágil inspirada no *extreme programming*. Os principais motivos para a escolha desta metodologia prendem-se com o facto de esta ser adequada para projetos de investigação onde se exploram tecnologias com as quais a equipa de desenvolvimento não está familiarizada. Por outro lado, esta metodologia permite ciclos de desenvolvimento mais curtos com recurso a melhorias iterativas, o que lhe permite ser mais tolerante a falhas ou alterações de rumo no projeto.

Ao longo do projeto foram sendo desenvolvidas várias *spikes* independentes que permitiram estudar e desenvolver soluções para problemas específicos, tais como:

- Ativação e interpolação dos modos de configuração e de reposição através do botão físico
- Deteção de movimentos com o *hardware*
- Comunicação sem fios com e sem ACK
- Definição do identificador e grupo para os nós
- Suspensão e ativação da interface de comunicação
- Persistência do estado e configurações em memória

De acordo com as necessidades, estas *spikes* foram depois integradas em cada um dos módulos no sentido da construção de um protótipo funcional.

Para este sistema foram desenvolvidos dois módulos distintos, um para implementação das funcionalidades do nó/caixa de correio (módulo 1) e outro para definição de um interface de comunicação de testes no sistema central (módulo 2).

5.2.1. Módulo 1 - nó/caixa de correio

O *software* do módulo 1 seria o responsável pela implementação dos algoritmos propostos na especificação. De acordo com estes algoritmos a principal função deste módulo seria a de enviar mensagens de estado para o sistema central, tendo ainda a capacidade de receber comandos de configuração.

No ficheiro `smartMailBox.ino` está definido o código-fonte responsável por estas instruções. No cabeçalho do ficheiro são incluídas as bibliotecas necessárias e definidas as estruturas de dados utilizadas na manutenção de estado. No bloco seguinte é definida a função responsável por persistir o estado do módulo em memória com recurso à memória disponível na EEPROM. São depois definidas mais algumas variáveis e estruturas como a utilizada na comunicação, constantes com os endereços botão, LED e PIR, etc. Nos blocos seguintes são definidas, respetivamente, as funções emissão do estado e apresentação na porta de série de alguns meta-dados relativos ao estado atual.

Na função `setup` são definidas as instruções de inicialização do *hardware* onde é inicializada a porta de série, o sensor PIR, o interface de comunicação e re posto o estado anterior (ou definido um estado por omissão). No final do ficheiro é definida a função `loop` com o código responsável por monitorizar o *hardware* e implementar os algoritmos propostos. Nas primeiras linhas é verificado o estado do botão de configuração, para reposição do estado sem correio e/ou para ativação do modo de configuração. No final desta função o sistema processa alterações de estado provocadas pela ativação do sensor PIR notificando o sistema central caso tenha sido detetado movimento.

5.2.2. Módulo 2 - sistema central

O módulo 2 foi desenvolvido como um sistema de testes para validar o funcionamento do módulo 1. Existem algumas semelhanças entre o código-fonte de

ambos os módulos, principalmente no que diz respeito à persistência das configurações e sistema de comunicação. A principal diferença deste módulo é a capacidade de este interpretar e executar comandos a partir da porta de série.

```
--cmdline: X

Available commands:
123 x      - Toggle configuration change protection, l=Unlocked
<nn> i     - set node ID (standard node ids are 1..26)
<n> b      - set MHz band (4 = 433, 8 = 868, 9 = 915)
<nnn> g     - set network group (RFM12 only allows 212, 0 = any)
<n> c      - set collect mode (advanced, normally 0)
...,<nn> a   - send data packet to node <nn>, with ack
...,<nn> s   - send data packet to node <nn>, no ack
----- Nodes configuration commands
<mm>,<nn> i - set new ID <mm> to node <nn>
<mm>,<nn> g - set new GROUP <mm> to node <nn>
<n> l       - turn activity LED on DIG8 on or off

All the commands should end with #

Current configuration:
79 i15 g0 @ 433 MHz Lock: 1

-
```

Imagen 11: Alguns dos comandos disponíveis no módulo do sistema central

Através deste módulo é possível configurar uma série de parâmetros, não só do sistema central, como o de qualquer nó compatível e em modo de configuração. Por exemplo, ao enviar o comando `14,10i#` estamos e dizer à central para enviar um comando de configuração para o nó identificado com o número 10 no grupo atual, para que este nó passe a assumir o identificador 14 no grupo atual. Do mesmo modo podemos enviar o comando `210,10g#` onde, neste caso, estamos a informar o sistema de que o nó identificado com o número 10 no grupo atual, deve passar a utilizar o grupo 210 para a comunicação.

Com este sistema é possível controlar o *hardware* de comunicação do sistema central a partir de qualquer *software* capaz de comunicar através da porta de série. Assim, foi também desenvolvido um pequeno *script* PHP que através dos dados especificados num formulário numa página web, foi capaz de configurar um nó através deste protocolo de comunicação.

6. Avaliação e análise

Após a implementação do protótipo este foi submetido a uma série de testes para avaliar a viabilidade da solução proposta.

```
Current configuration:
```

```
i10 g210 @ 433 MHz hasMail 0
calibrating sensor ..... done
SENSOR ACTIVE
Setup done!
---
motion detected at 37 sec

motion ended at 40 sec
Notify
  hasMail: 1
  battery: 100
```

Imagen 12: Registo da informação obtida num dos testes efetuados

Na imagem 12 o nó da caixa de correio começa por carregar e repor o seu estado. Neste caso o nó tem o identificador 10 no grupo 210, está a operar a uma frequência para comunicação de 433MHz e não está registada a presença de correio. Após a calibração inicial do sensor, o nó está pronto a detetar movimento; é o que acontece 37 segundos após a sua inicialização pela introdução de correspondência na caixa. Após se detetar que não existe mais movimento, é então enviada uma notificação a indicar a presença de correio e o estado da bateria.

A solução foi submetida a uma série de testes tais como a introdução de vários tipos de materiais com diversos tamanhos e espessuras, deteção desses mesmos objetos com várias intensidades de iluminação (desde escuridão absoluta até à iluminação artificial intensa). Apesar de não numa forma tão exaustiva, foram ainda executados alguns testes de distância entre o nó e o sistema central com e sem barreiras que pudesse atenuar o sinal. Consegiu-se obter comunicação até uma

distância de aproximadamente 30 metros e, embora não se tenha testado com distâncias superiores, nas páginas dos fornecedores são referidas distâncias superiores a 150 metros para o *hardware* utilizado², dependendo este alcance da qualidade da antena utilizada.

² http://embeddedwirelessolutions.com/rfm12b_433mhz_transceiver

7. Conclusões e trabalhos futuros

Pelos testes realizados é possível concluir que a solução proposta é funcional e tecnologicamente viável. Verificou-se que foi possível detetar movimento com origem na introdução de correspondência numa caixa de correio e enviar uma notificação através de um meio sem fios para um sistema central.

Como trabalhos futuros sugere-se a execução de testes adicionais que permitam validar e otimizar a distância máxima da comunicação entre o nó e o sistema central, testando a influência de interferências físicas e concorrência com outros nós na comunicação. Do ponto de vista do código com as instruções de controlo da solução, este pode ser melhorado através a inserção de otimizações e integração com outros sistemas inteligentes já disponíveis no mercado.

Este foi um projeto desafiante que permitiu o contacto com tecnologias com as quais não estávamos tão familiarizados e aplicação de conceitos associados ao desenvolvimento de objetos inteligentes.

Referências

- [1] United States Postal Office, *Postal Facts 2013*, Corporate Communications, 2013 <<https://about.usps.com/who-we-are/postal-facts/welcome.htm>> [Acedido a 12 de Abril de 2012].
- [2] Wikipedia contributors, *Ubiquitous computing*, Wikipedia, The Free Encyclopedia, 7 de Abril 2013 <http://en.wikipedia.org/w/index.php?title=Ubiquitous_computing&oldid=549127642> [Acedido a 12 de Abril de 2013].
- [3] Wikipedia contributors, *Smart objects*, Wikipedia, The Free Encyclopedia, 20 de Fevereiro 2013 <http://en.wikipedia.org/w/index.php?title=Smart_objects&oldid=539214825> [Acedido a 12 de Abril de 2013].
- [4] Christopher Awobue, *Smart mailbox - US 7256691 B2*, Awobue Christopher, 14 de Agosto 2007 <<http://www.google.com/patents/US7256691>> [Acedido a 12 de Abril de 2013].
- [5] Melvin Adams, *Remote mail delivery indicator system*, Melvin Adams, 14 de Setembro 1999 <<http://www.google.com/patents/US5950919>> [Acedido a 12 de Abril de 2013].
- [6] Peter Staples, *System to detect mail in a mailbox*, Blue Clover Design, Llc, 9 de Junho 2005 <<http://www.google.com/patents/US20050122220>> [Acedido a 12 de Abril de 2013].
- [7] Mike Riley, *Programming Your Home*, The Pragmatic Programmers, 2012.
- [8] STI, *Installation and Operation Manual*, Safety Technology International, 2012.