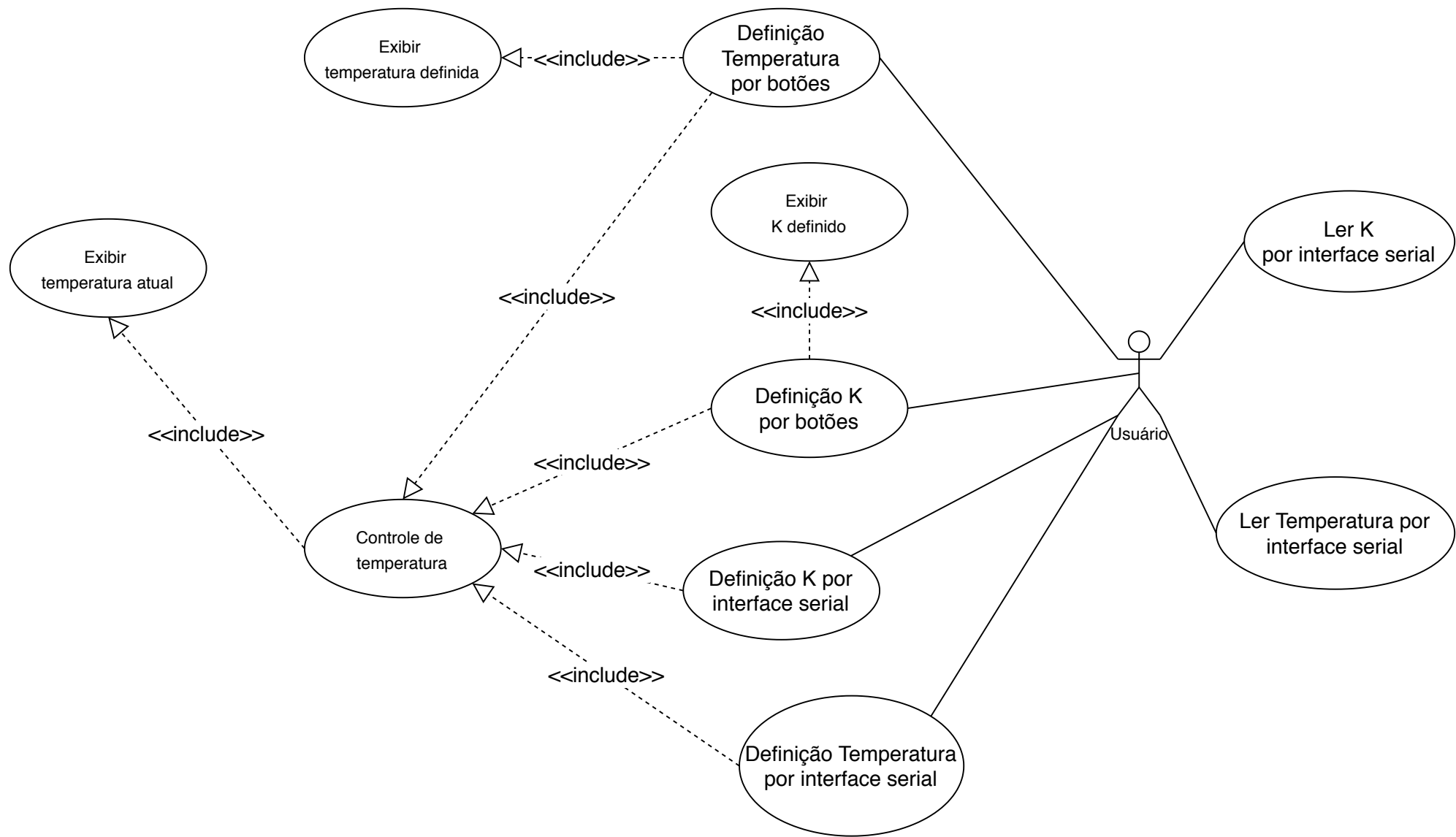


1. Tabela de requisitos e especificações do sistema

Com base nas descrições do sistema foi levantada uma tabela de requisitos funcionais e não funcionais, listando exigências do projeto. Também foi construída uma tabela com as especificações de materiais para construção do projeto, como utilizamos o kit de desenvolvimento McLab2 recomendamos a utilização do mesmo para obter desempenhos semelhantes.

2.1 Diagrama de casos de uso

Levando em conta as ações que o usuário aplica no sistema foi construído o seguinte diagrama de caso de uso.



Obs.:

O caso de exibir temperatura atual, deve ocorrer sempre que uma interrupção é finalizada, a fim de que sempre esteja disponível a temperatura atual para a visualização do usuário.

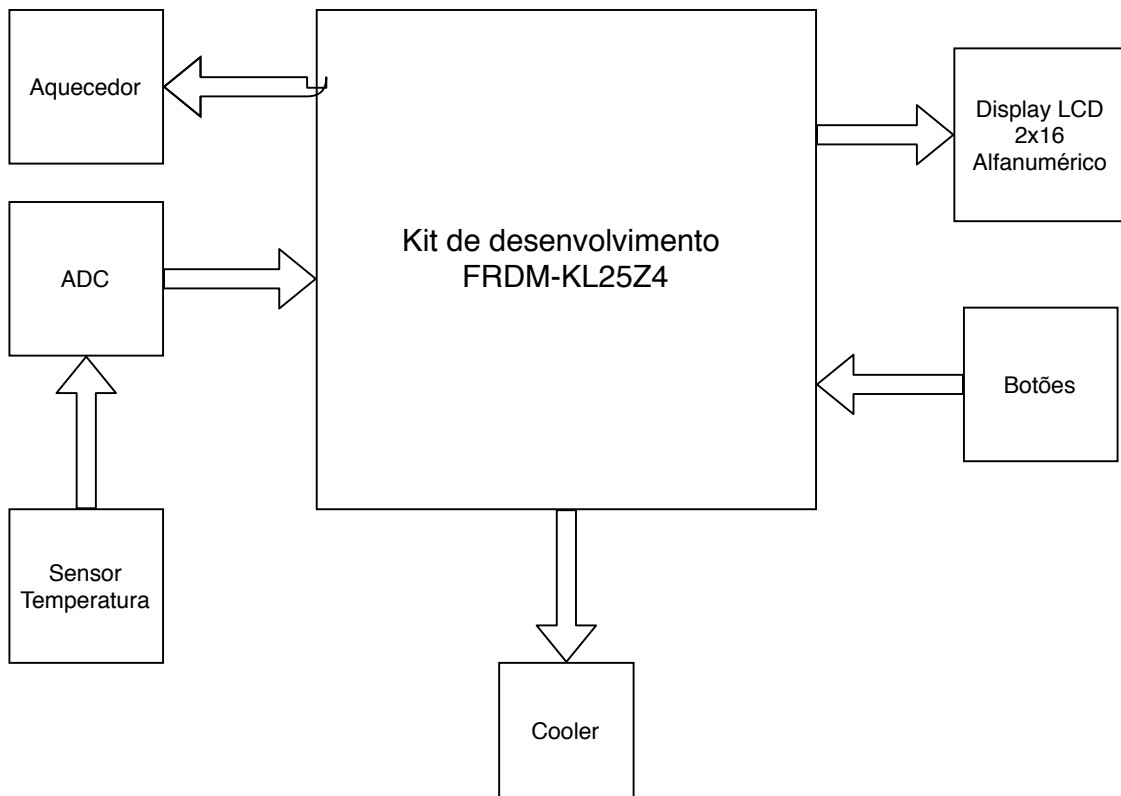
2.2 Diagrama de casos de cenário

Com base no diagrama de caso de usos, foram construídas 6 tabelas representando o cenário proposto para cada caso. Um caso para definir temperatura por botões, outro por comunicação serial, um para definir os ganhos por botões, outros 3 por comunicação serial, e um para cada get pela serial.

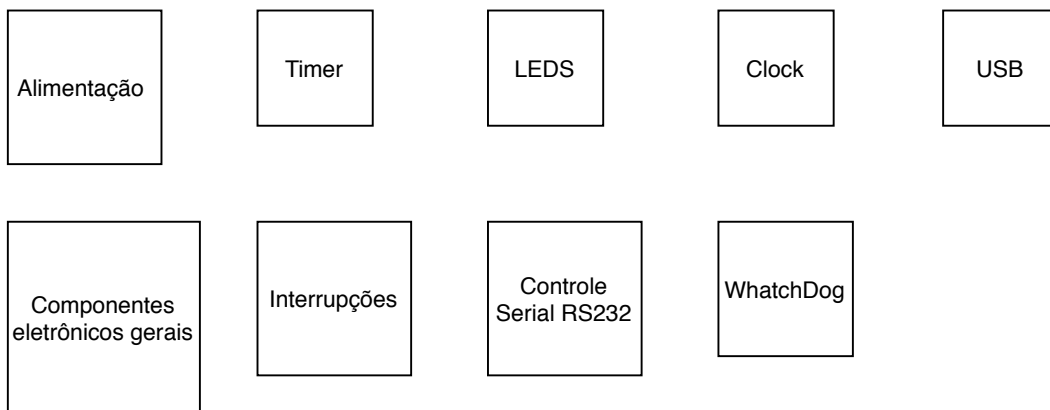
Caso de uso: Definir temperatura por botões				Caso de uso: Definir temperatura por serial				Caso de uso: Ler temperatura por serial			
Atores: Usuário.				Atores: Usuário.				Atores: Usuário			
Pré-requisito: O usuário deve acionar a interrupção pressionando o botão 1.				Pré-requisito: O usuário deve enviar um comando por interface serial RS-232.				Pré-requisito: O usuário deve enviar um comando por interface serial RS-232.			
Descrição: Definir uma temperatura para o controle do sistema.				Descrição: Definir uma temperatura para o controle do sistema.				Descrição: Ler o valor de temperatura através da interface serial RS-232.			
Fluxo de Eventos Principais:				Fluxo de Eventos Principais:				Fluxo de Eventos Principais:			
1. Acionar a interrupção do botão correspondente à temperatura;				1. Verificar se o comando é de set ou get;				1. Verificar se o comando é de set ou get;			
2. Definir a temperatura desejada para o sistema;				2. Se for de set, verificar se a variável a ser setada é o 't' de temperatura;				2. Se for de get, verifica a variável desejada;			
3. Se ele pressionar o botão 2 aumenta a temperatura em 1°C;				3. Se a variável for 't', verifica o valor desejado para a definição				3. Se for 't', lê o valor de temperatura e atualiza resposta;			
4. Se ele pressionar o botão 3 aumenta a temperatura em 10° C;				4. Aguarda a interrupção do controle para atualizar a temperatura e exibir no LCD;				4. Envia a resposta através da interface serial;			
5. Se ele pressionar o botão 1 confirma a temperatura setada;								5. Aguarda a interrupção do controle para atualizar a temperatura e exibir no LCD;			
7. Aguarda a interrupção do controle para atualizar a temperatura e exibir;											
Caso de uso: Definir ganhos por botões				Caso de uso: Definir ganhos por serial				Caso de uso: Ler ganhos por serial			
Atores: Usuário				Atores: Usuário				Atores: Usuário			
Pré-requisito: O usuário deve acionar a interrupção pressionando o botão 2				Pré-requisito: O usuário deve enviar um comando por interface serial RS-232.				Pré-requisito: O usuário deve enviar um comando por interface serial RS-232.			
Descrição: Definir os três ganhos para o controle do sistema				Descrição: Definir ganhos para o controle do sistema.				Descrição: Ler o valor dos ganhos através da interface serial RS-232.			
Fluxo de Eventos Principais:				Fluxo de Eventos Principais:				Fluxo de Eventos Principais:			
1. Acionar a interrupção do botão correspondente aos controladores				1. Verificar se o comando é de set ou get;				1. Verificar se o comando é de set ou get;			
2. Definir um controlador por vez				2. Se for de set, verifica a variável a ser setada;				2. Se for de get, verifica a variável a ser lida;			
3. Define e exibe Kp				3. Se for 'p', verifica o valor desejado e atualiza o Kp;				3. Se for 'p', lê o valor de Kp e atualiza a resposta;			
4. Se ele pressionar o botão 2 aumenta o K em 1°C;				4. Se for 'i', verifica o valor desejado e atualiza o Ki;				4. Se for 'i', lê o valor de Ki e atualiza a resposta;			
5. Se ele pressionar o botão 3 aumenta o K em 10°C;				5. Se for 'd', verifica o valor desejado e atualiza o Kd;				5. Se for 'd', lê o valor de Kd e atualiza a resposta;			
6. Se ele pressionar o botão 1 confirma o K setado;				6. Aguarda a interrupção do controle para atualizar a temperatura e exibir no LCD;				6. Envia a resposta através da interface serial;			
7. Repetir passos 3 a 6 para Ki								7. Aguarda a interrupção do controle para atualizar a temperatura e exibir no LCD;			
8. Repetir passos 3 a 6 para Kd											
9. Se ele pressionar o botão 1 atualiza todos os ganhos no sistema;											
10. Aguarda a interrupção do controle para atualizar a temperatura e exibir;											

3 Diagrama em blocos do Hardware

Analizando a estrutura do nosso projeto, construímos o diagrama em blocos a seguir, representando as conexões de hardware e os detalhes da placa FRDM-KL24.

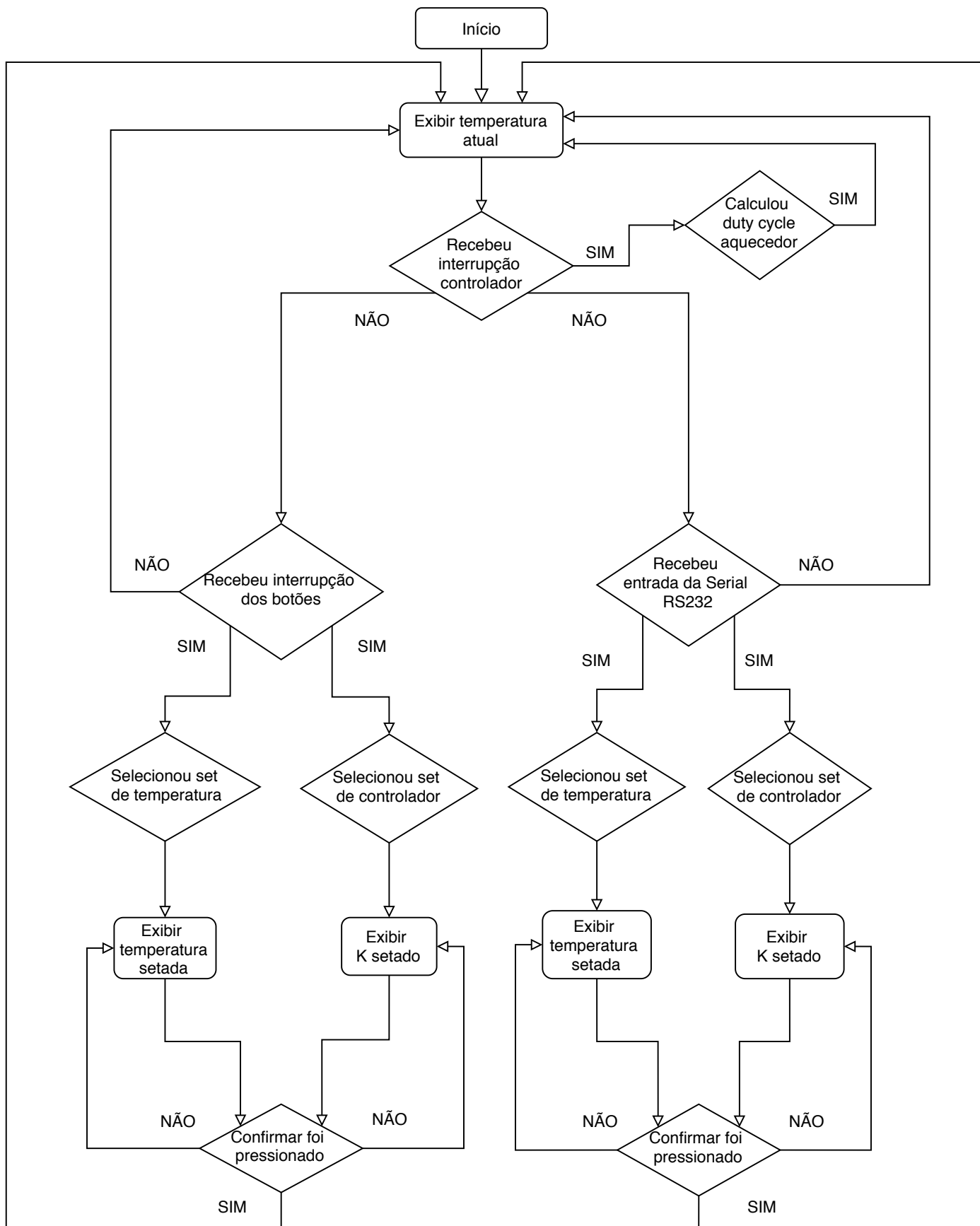


Kit FRDM-KL24



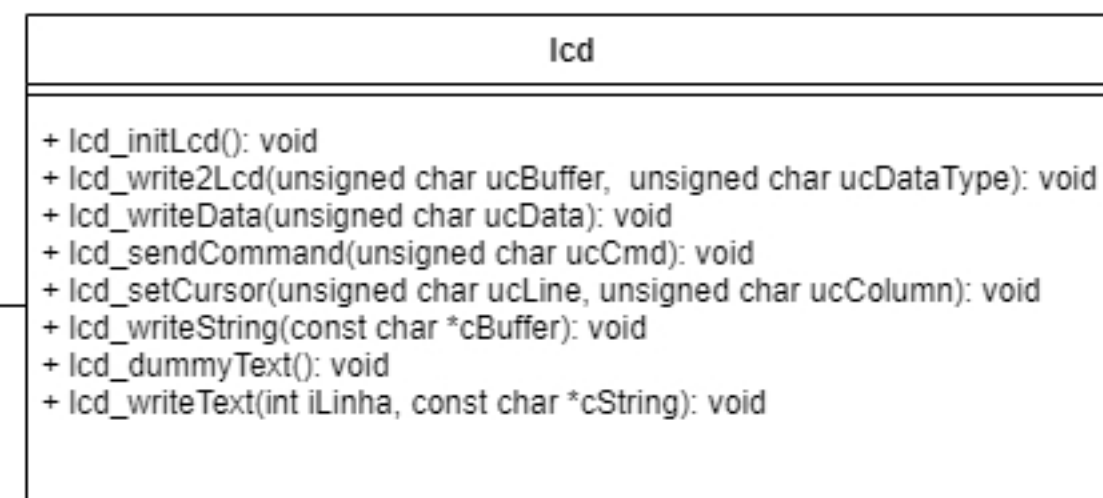
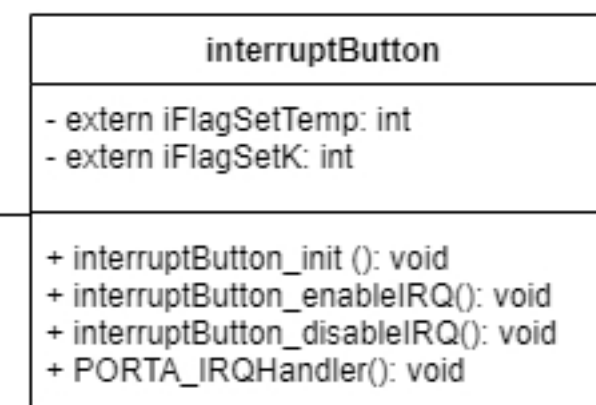
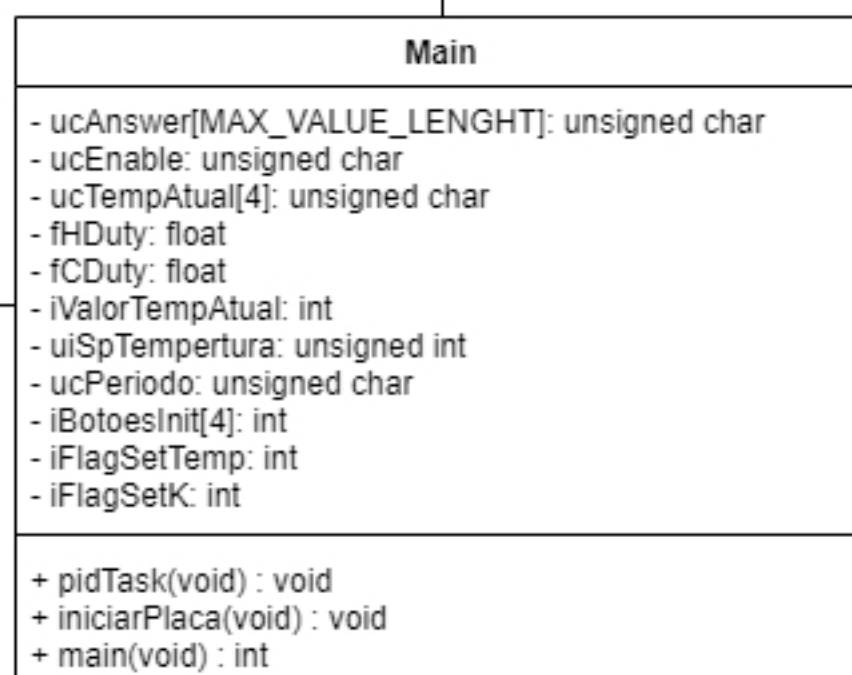
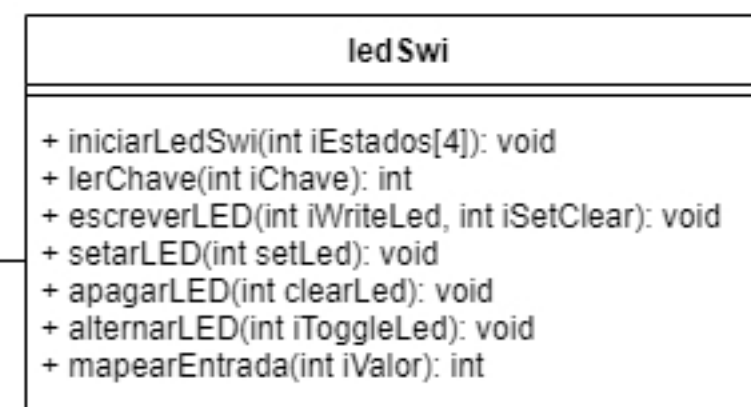
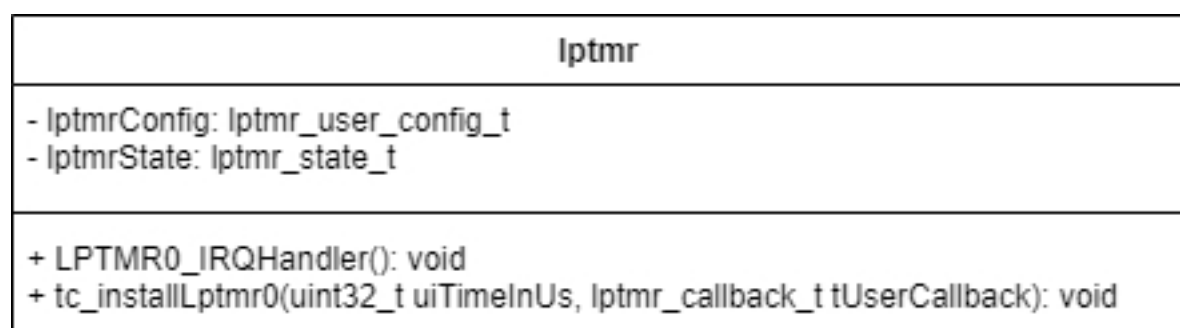
4 Fluxograma do algoritmo

Construídos o fluxograma a seguir para representar o nosso algoritmo, não foram representadas todas as verificações de interrupção no meio do fluxo a fim de simplificar o diagrama.



5 Diagrama UML de Classes

Analizando nosso código, construímos o diagrama de classes, tentando representá-lo pelo paradigma de orientado a objetos.

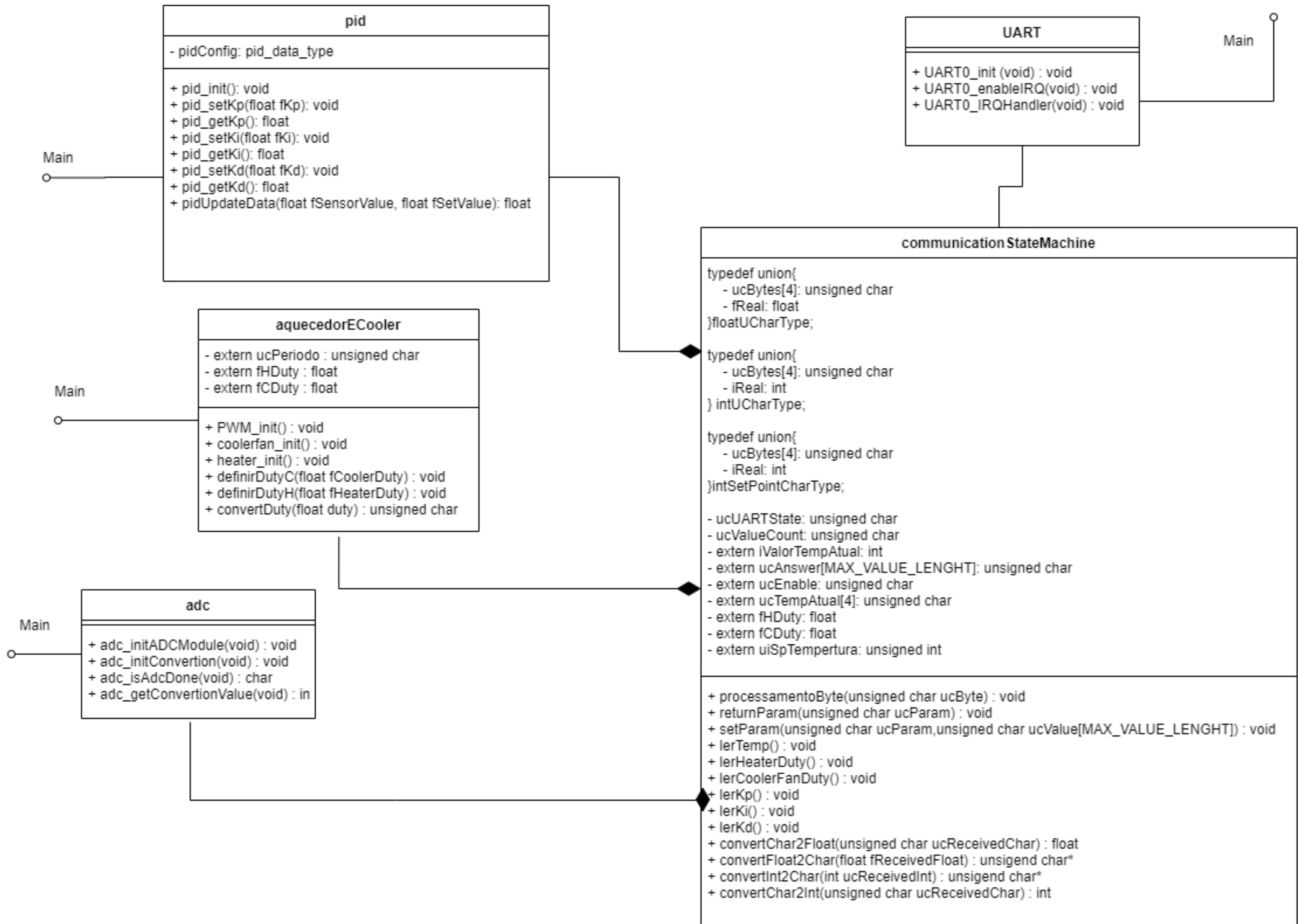


pid

aquecedorECooler

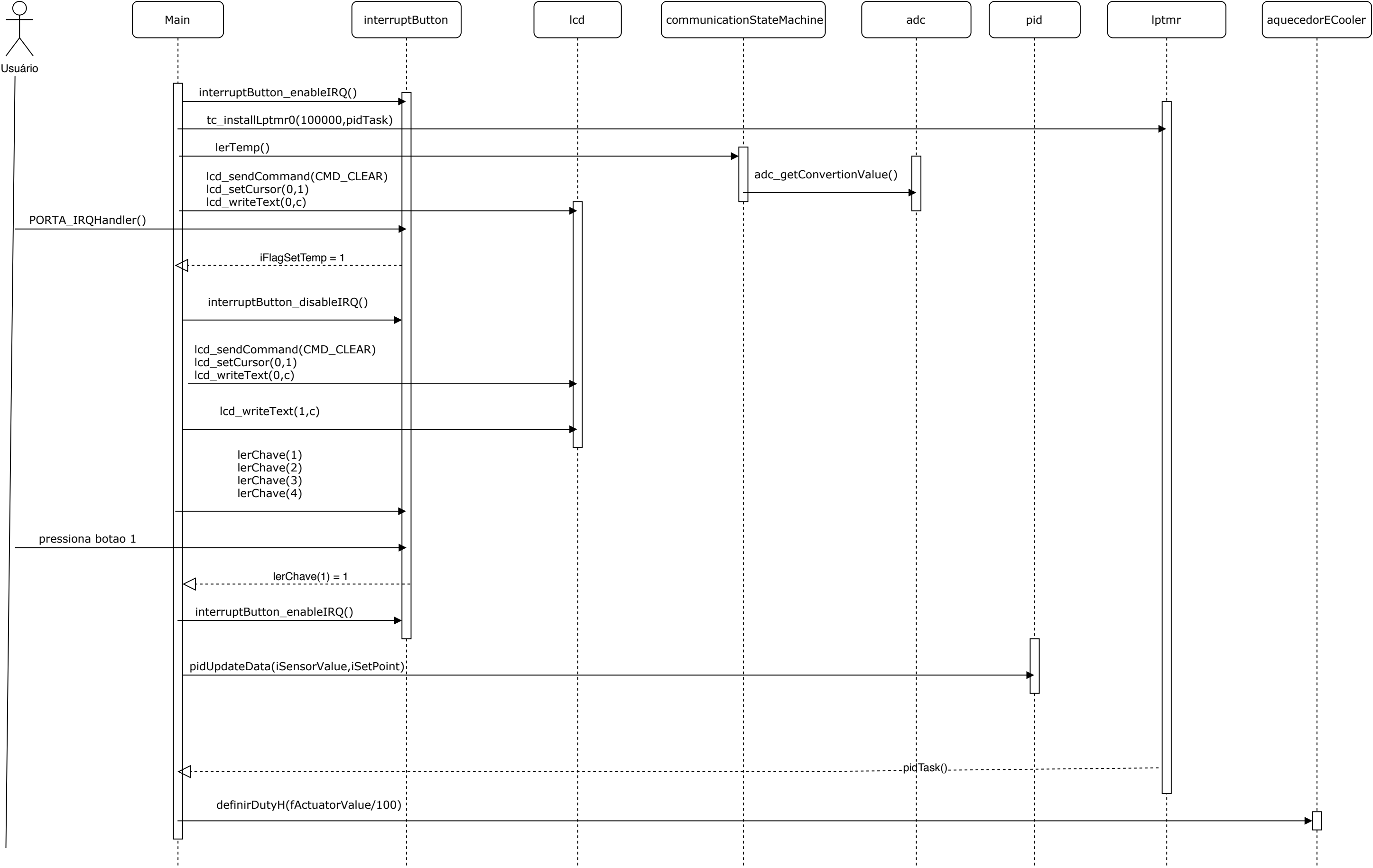
adc

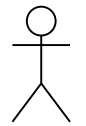
UART



6 Diagrama UML de Sequência

Com base no diagrama de classes e no diagrama de casos de uso, foram construídos 6 diagramas de sequência, um para cada caso de uso como no diagrama de cenários.





Usuário

Main

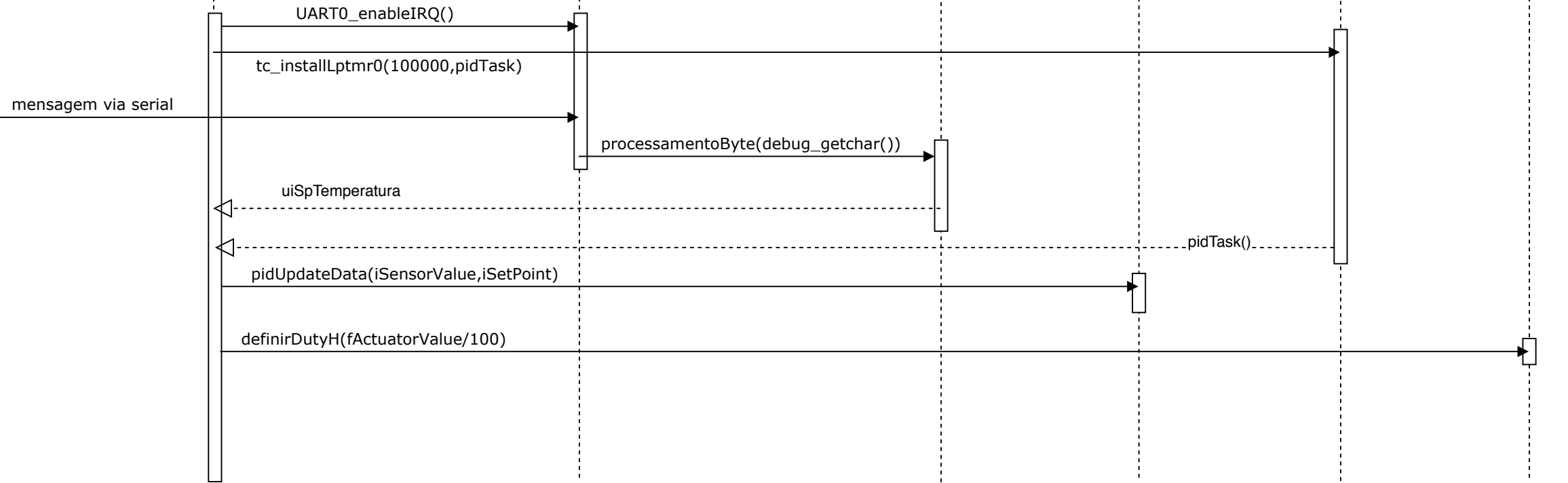
UART

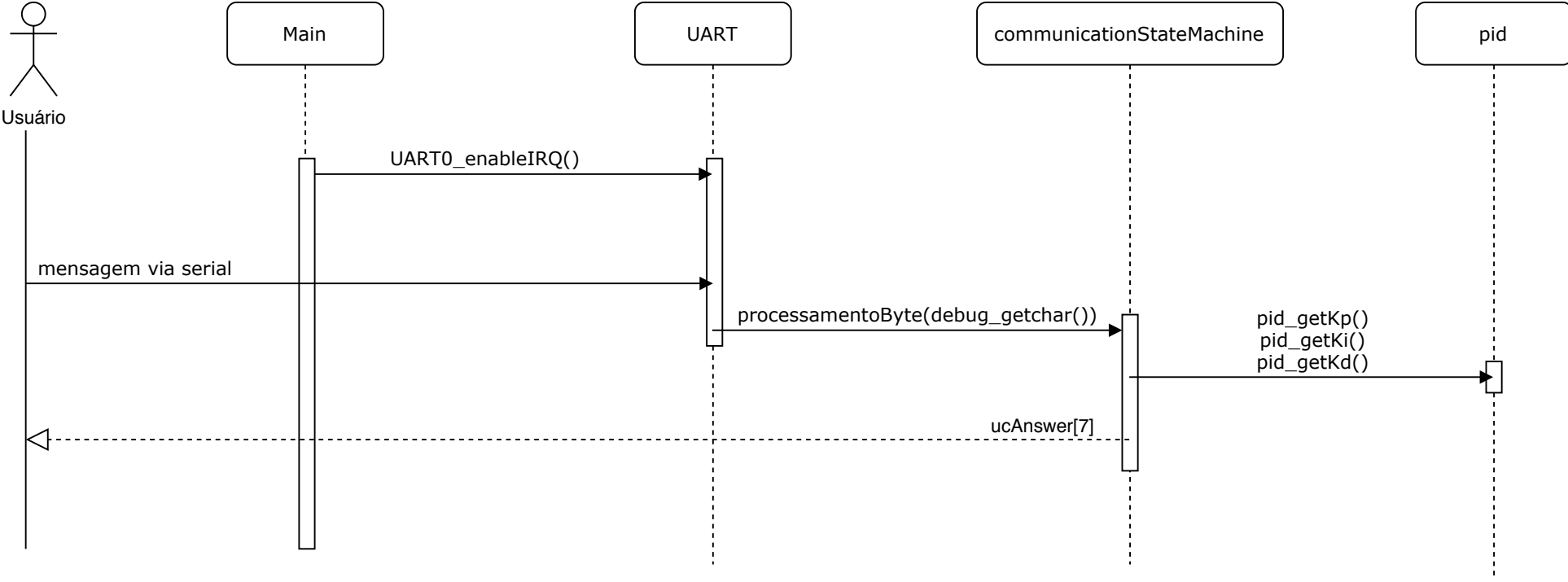
communicationStateMachine

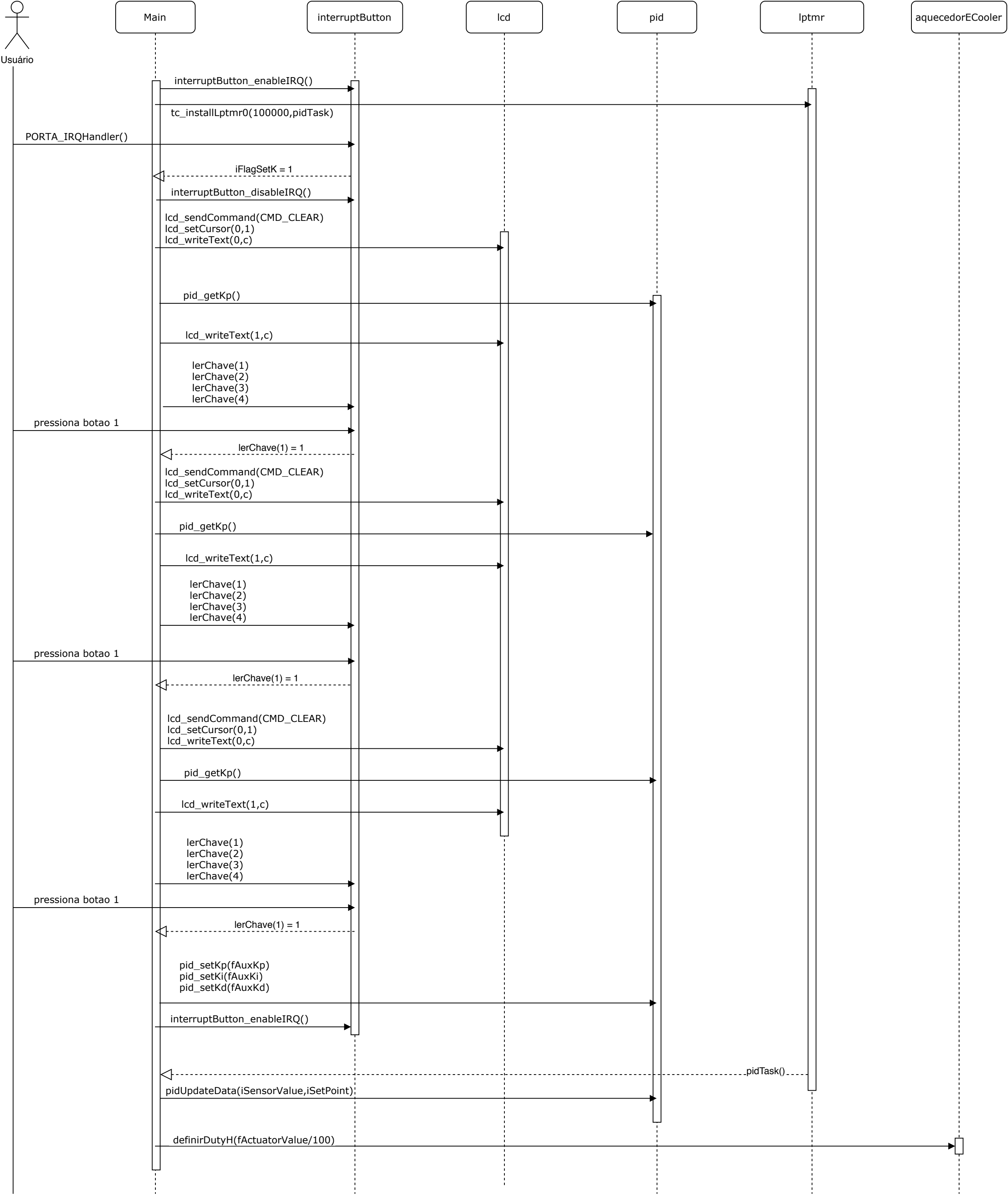
pid

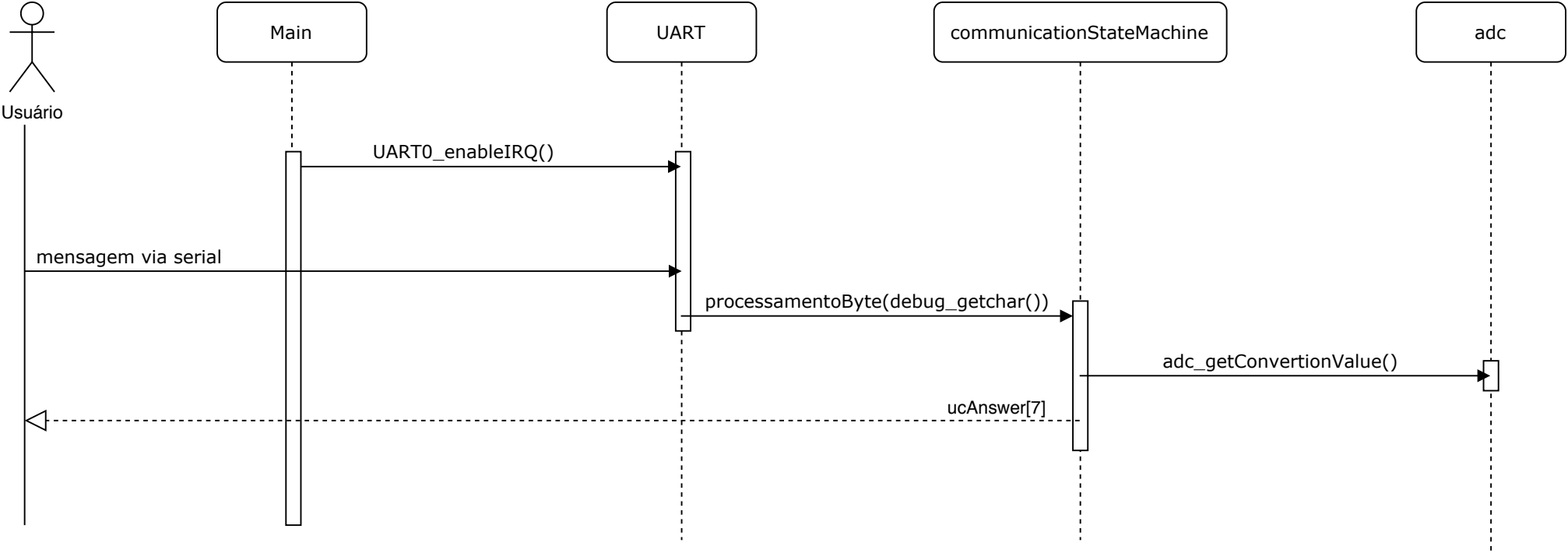
lptmr

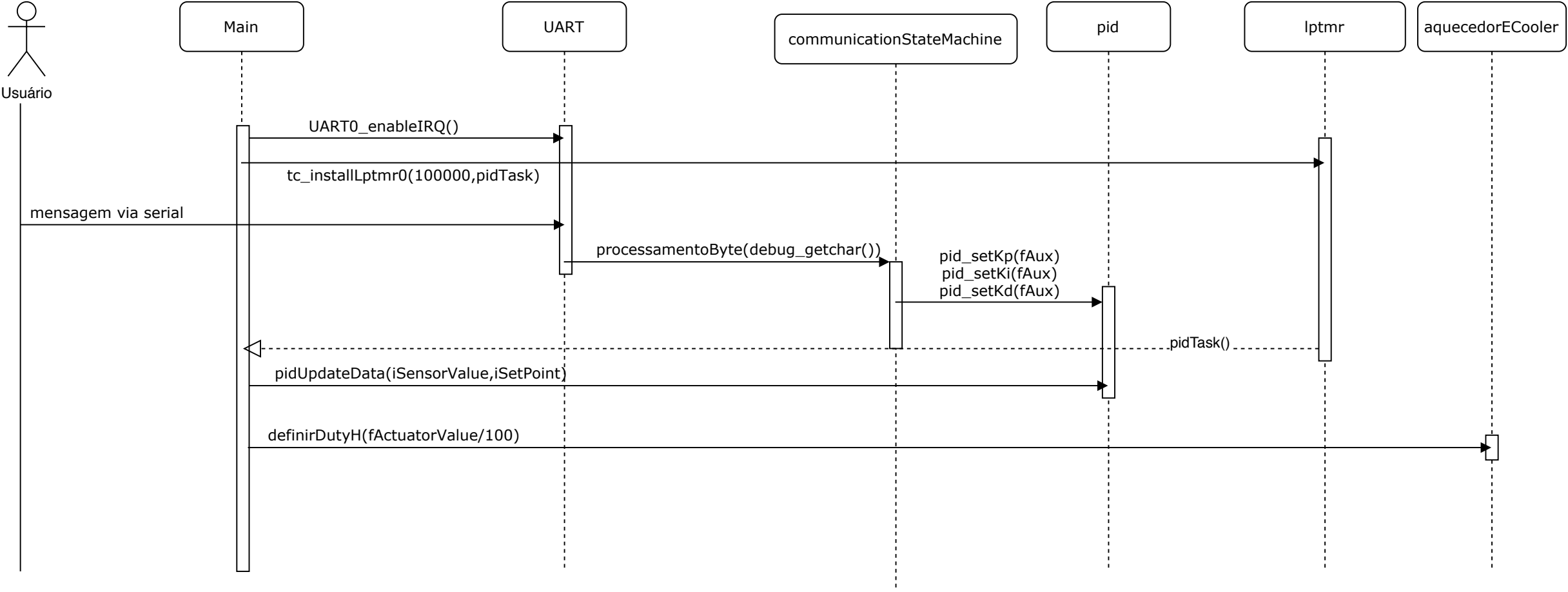
aquecedorECooler





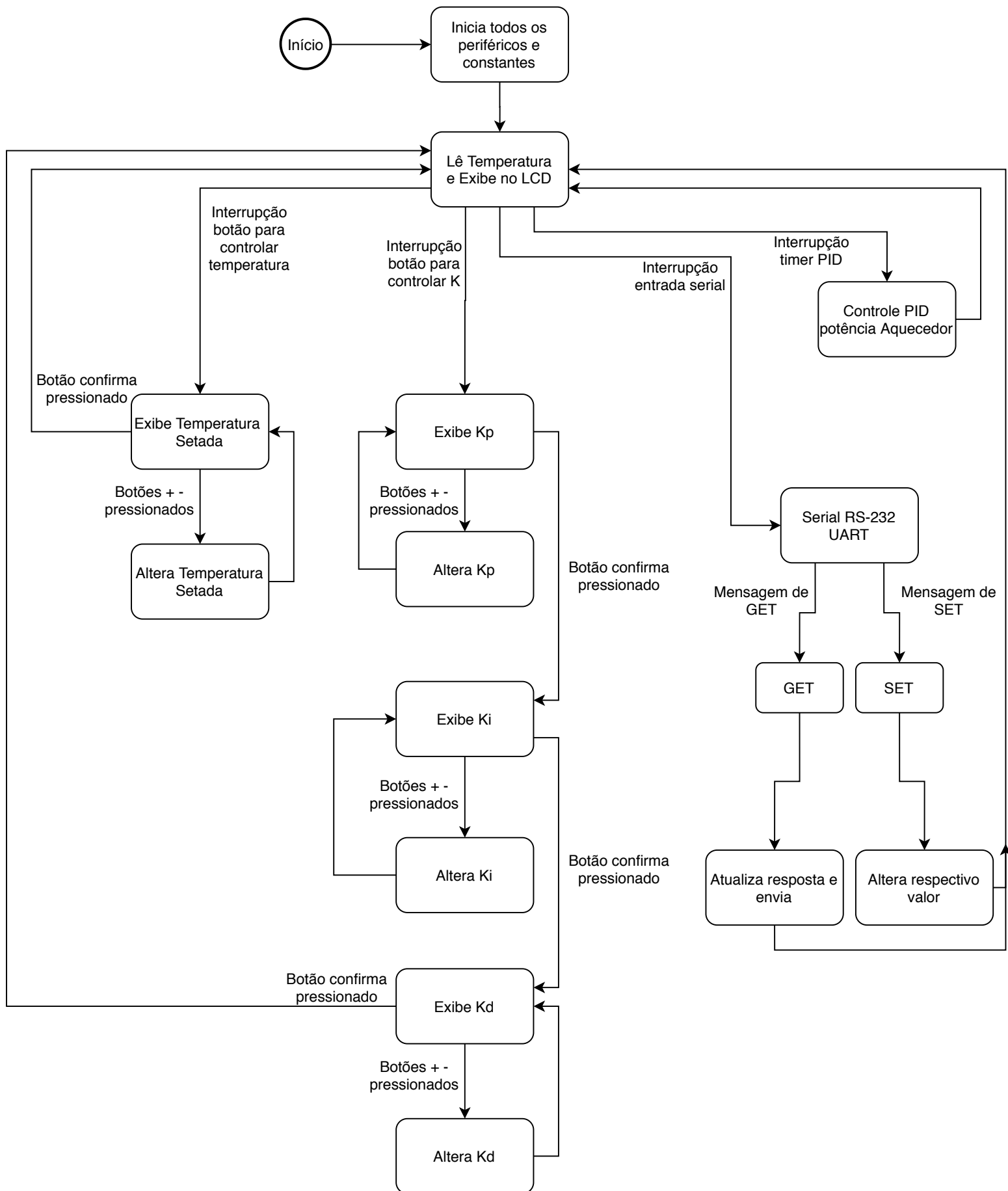






7 Diagrama UML de Máquina de Estados

A máquina de estados foi elaborada levando em conta quais estados o sistema terá em cada instante do programa, também não foram representadas as interrupções do controlador, pois elas podem ocorrer em qualquer estado.



Obs.:

Em todos os estados deve-se verificar se a interrupção do timer do PID foi adicionada e trata-la, a fim de não perder a periodicidade de 100ms que o controlador precisa para funcionar. Para simplificar a máquina de estados e evitar a poluição visual, essa verificação foi removida do diagrama, mas está acontecendo no sistema.