

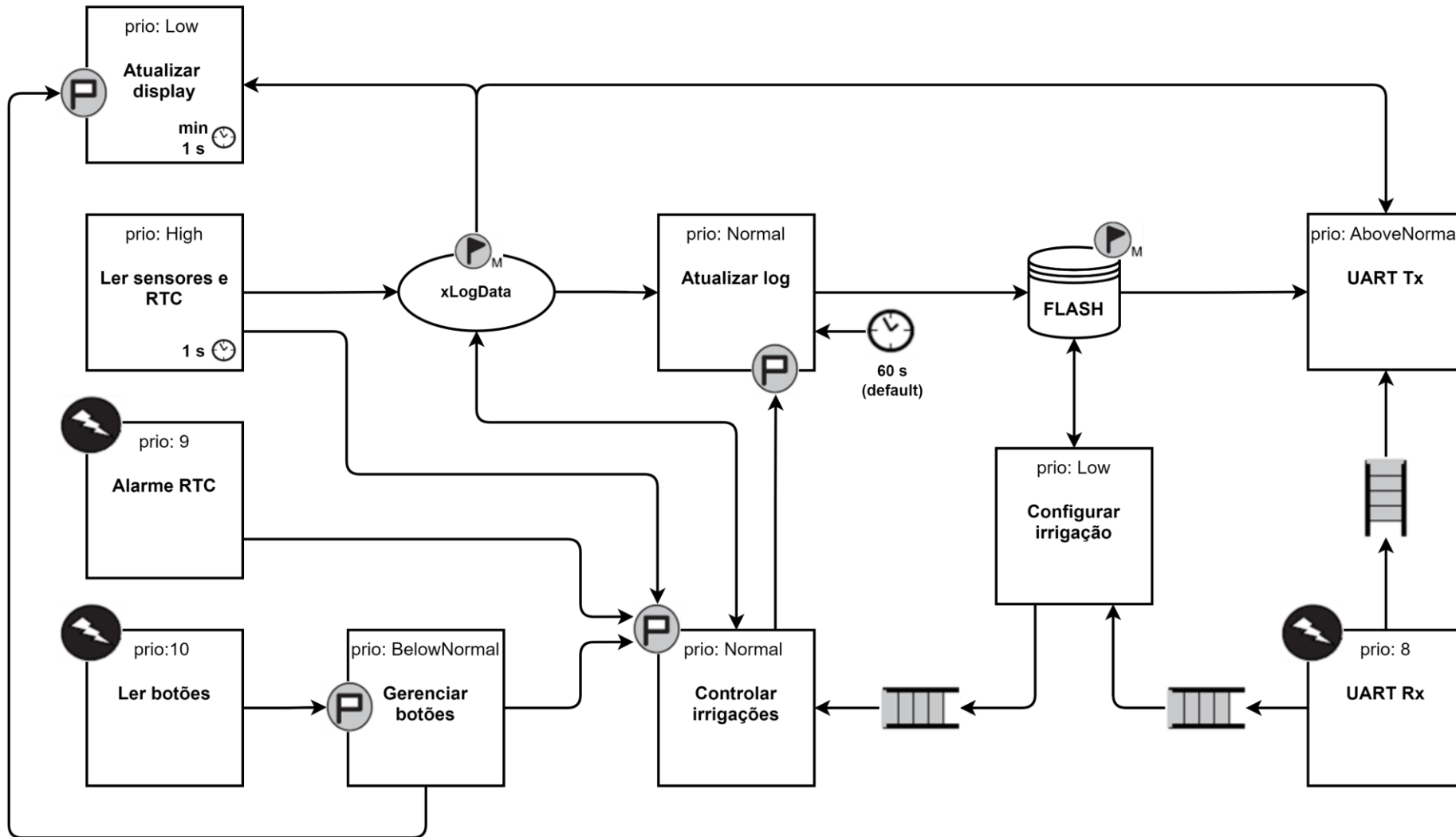


Projeto REGGAE

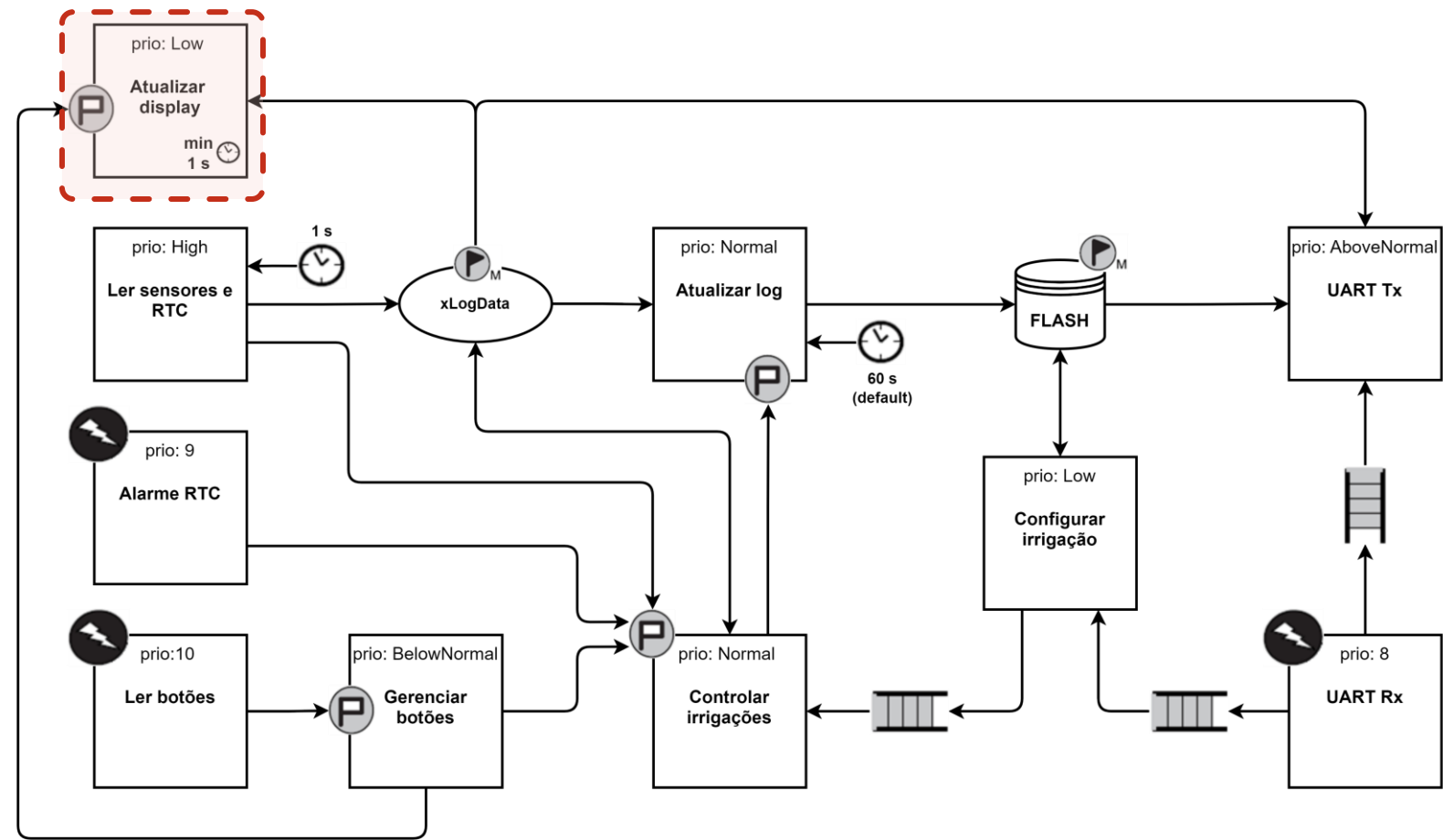
Etapas 2 – Projeto de software

RTOS-based
Environmental data
Gathering &
Garden
Automation
Experiment

Componentes RTOS

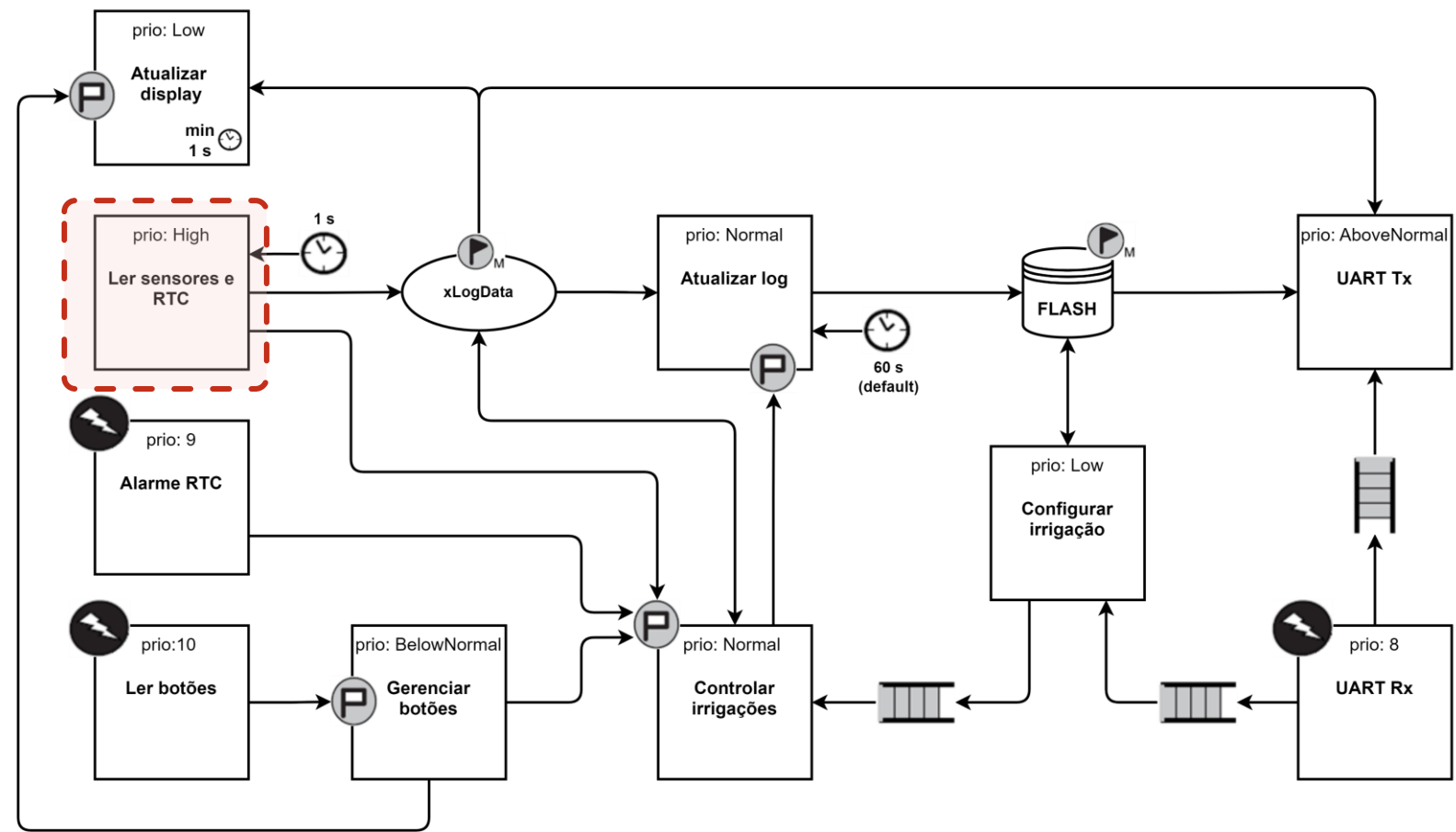


Componentes RTOS | *Tarefas*



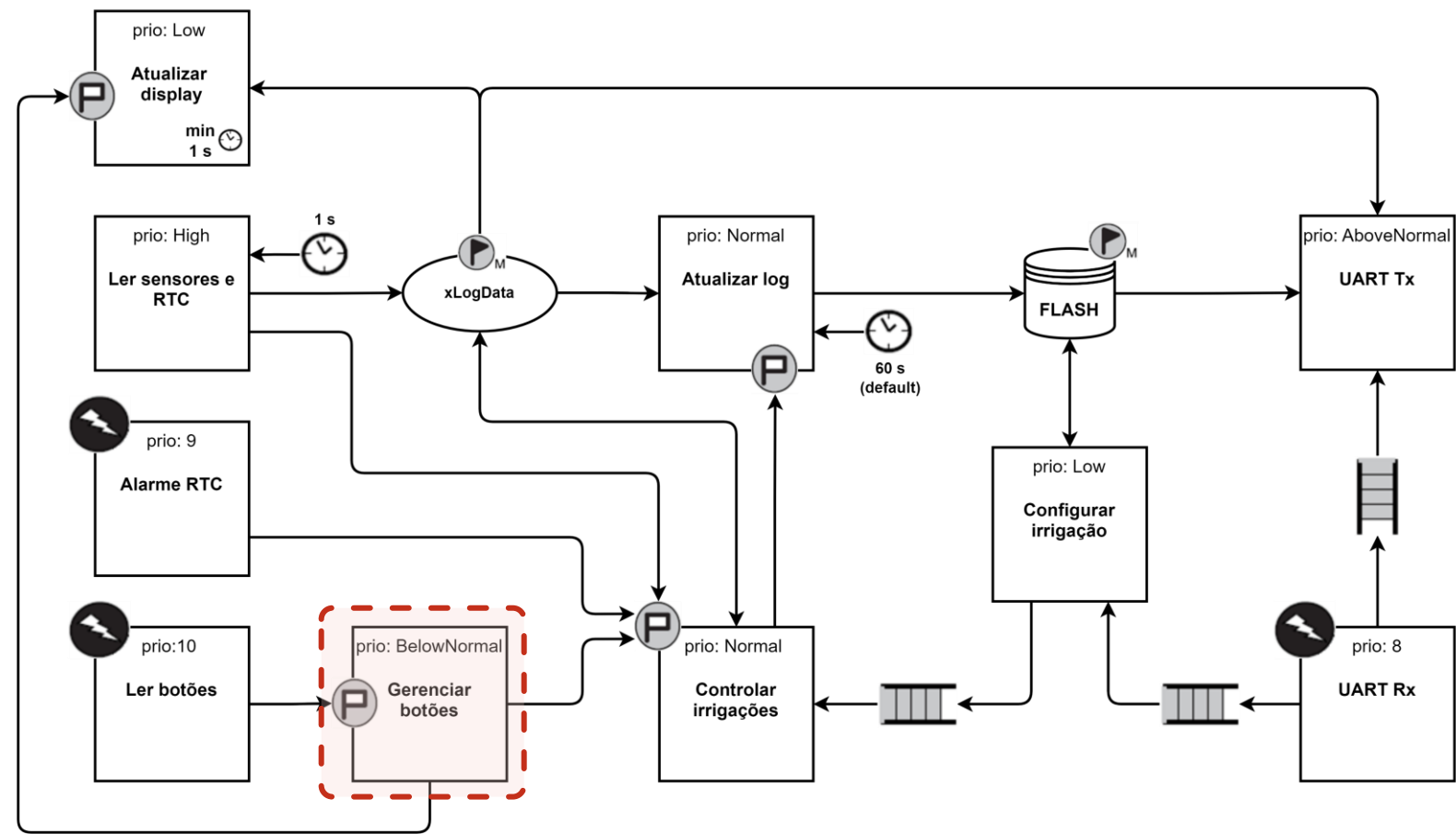
Tarefa: Atualizar display	
Nome	updateDisplay
Prioridade	Low
Stack	2 KB
Detalhes	Inicializa e atualiza exibição de dados no display via interface I2C. Os dados exibidos são horário, medidas de umidade do solo, temperatura e umidade ambiente e status das irrigações, obtidas da variável global <i>xLogData</i> , acessada via mutex. Usa um <i>osDelay</i> para atualizar o display a cada 1 segundo, no mínimo. Exibe canal de irrigação selecionado por botões, atualizado via <i>thread flag</i> pela tarefa <i>manageButtons</i> .

Componentes RTOS | *Tarefas*



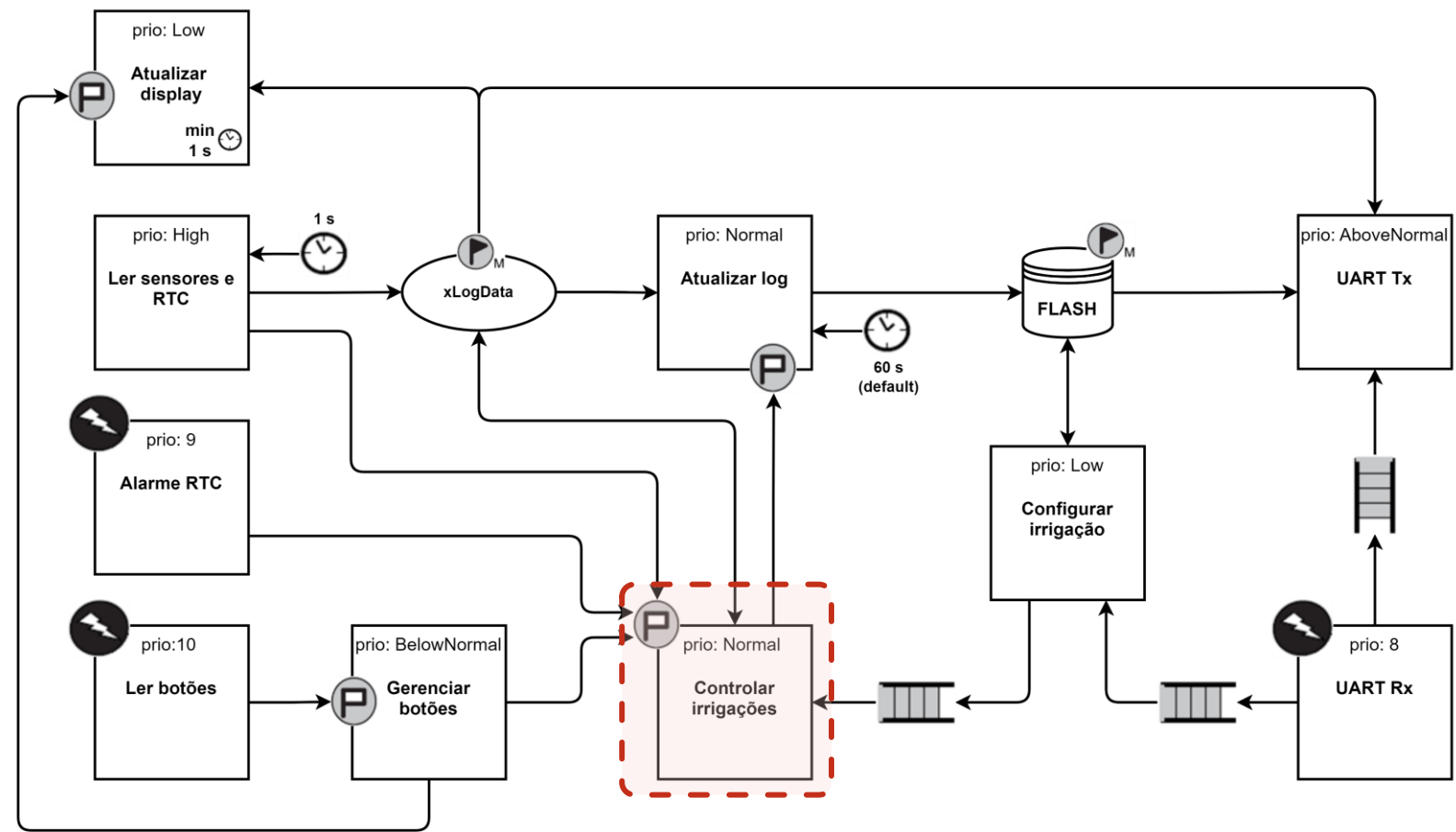
Tarefa: Ler Sensores e RTC	
Nome	readSensorsRTC
Prioridade	High
Stack	200 Bytes
Detalhes	Realiza a leitura dos 5 conversores A/D correspondentes às medidas de umidade de solo, do sensor DHT para umidade de temperatura ambiente e lê o valor de data e hora (dia, mês, ano, hora, minuto e segundo) do RTC. Escreve essas leituras na variável global <i>xLogData</i> , via mutex e sinaliza tarefa <i>controlWatering</i> com <i>thread flag</i> . É executada a cada 1 segundo, temporizada por um software timer.

Componentes RTOS | *Tarefas*



Tarefa: Gerenciar botões	
Nome	manageButtons
Prioridade	BelowNormal
Stack	100 Bytes
Detalhes	Executa lógica de acionamento manual da irrigação a partir <i>thread flags</i> geradas pela ISR de dois botões: um para escolher o canal (<i>selectChannel</i>), e outro para alternar o estado do canal (<i>toggleChannelState</i>). Através de <i>thread flags</i> , sinaliza a tarefa <i>updateDisplay</i> qual canal está selecionado, e a tarefa <i>controlWatering</i> qual canal deve alternar o estado.

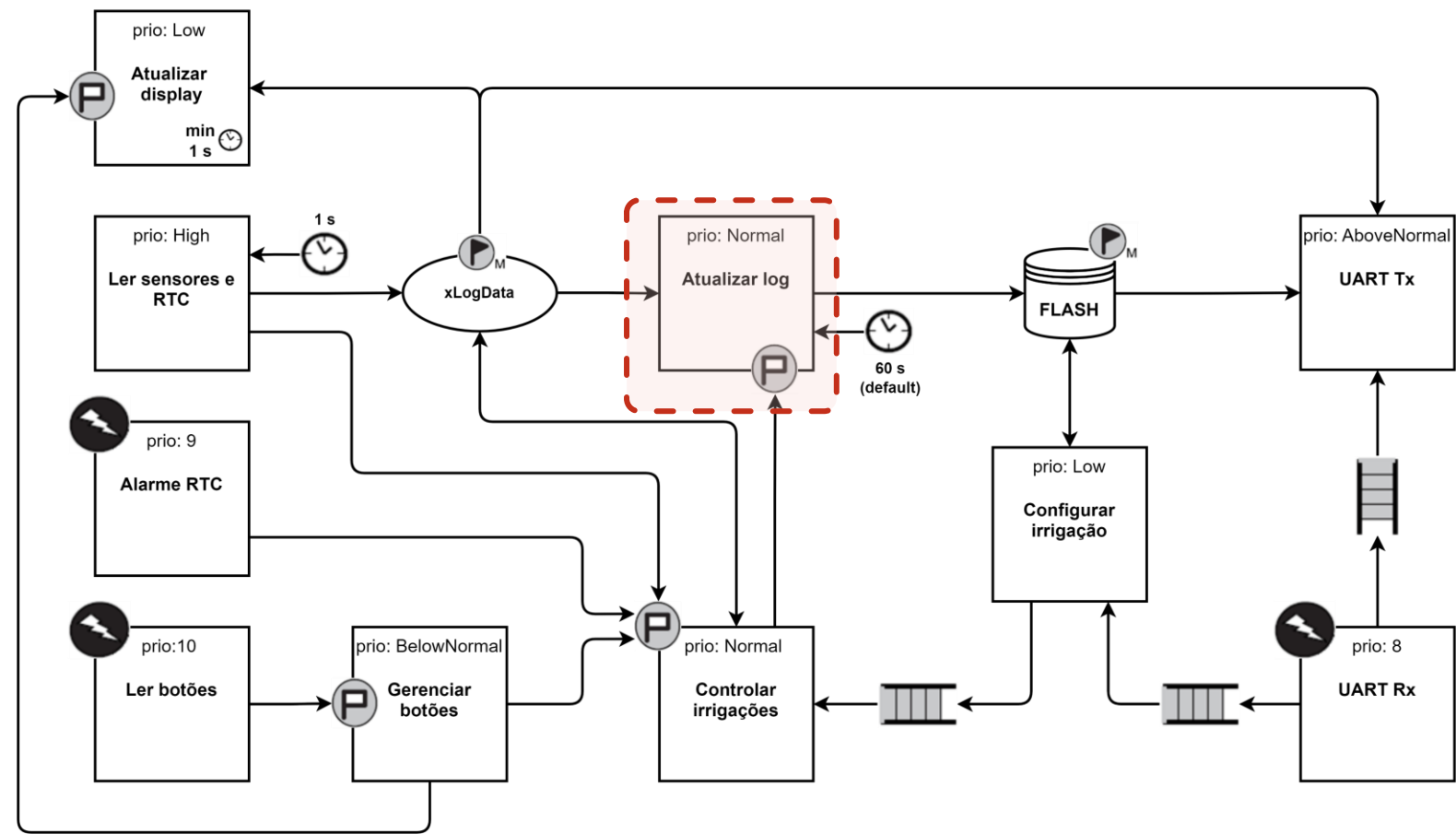
Componentes RTOS | *Tarefas*



Tarefa: *Controlar irrigações*

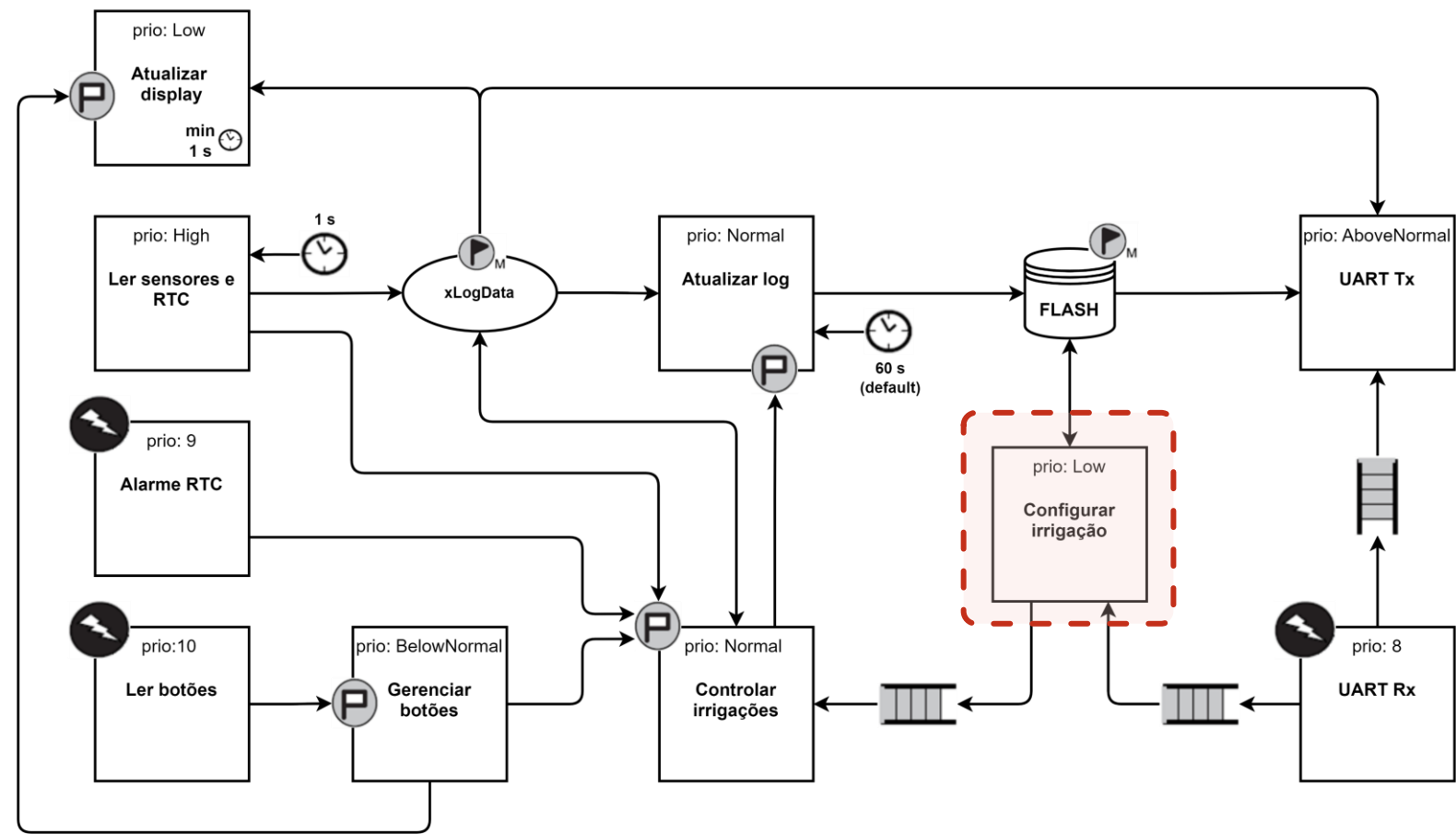
Nome	controlWatering
Prioridade	Normal
Stack	1 KB
Detalhes	Ativa e desativa GPIO's que acionam relés das válvulas de irrigação a partir das regras automática (periódica ou nível de umidade do solo) ou botões. Regras são configuradas via fila de mensagem recebida da tarefa <i>configureWatering</i> . Através de <i>thread flags</i> , é sinalizada pela ISR <i>alarmRTC</i> e tarefas <i>readSensorsRTC</i> e <i>manageButtons</i> para executar a lógica das regras de irrigação. Através de mutex, lê umidades dos solos e escreve status das irrigações na variável global <i>xLogData</i> . Sinaliza tarefa <i>updateLog</i> quando alguma irrigação for acionada ou desacionada. Reprograma alarmes do RTC para próximo acionamento periódico.

Componentes RTOS | *Tarefas*



Tarefa: Atualizar log	
Nome	updateLog
Prioridade	Normal
Stack	200 Bytes
Detalhes	Temporizado por um software timer configurável (com default de 60 segundos), escreve em buffer circular na FLASH uma nova aquisição, composta por: data, hora, umidade dos solos, temperatura e umidade do ar e status das irrigações (ligadas/desligadas). Através de <i>thread flag</i> , é sinalizada para escrever uma entrada “assíncrona” disparada por um acionamento de irrigação, seja ele automático ou manual.

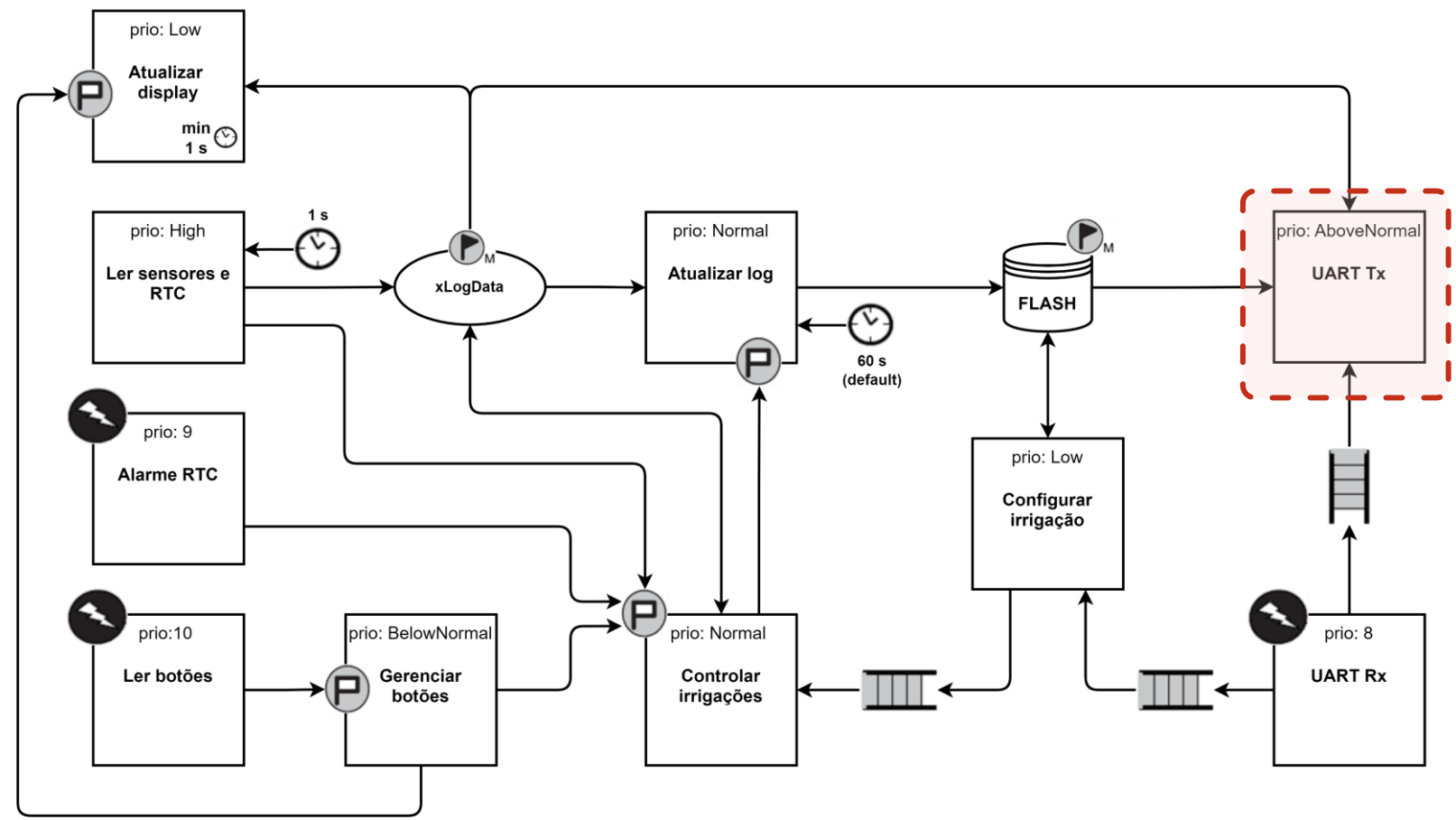
Componentes RTOS | *Tarefas*



Tarefa: Configurar irrigação

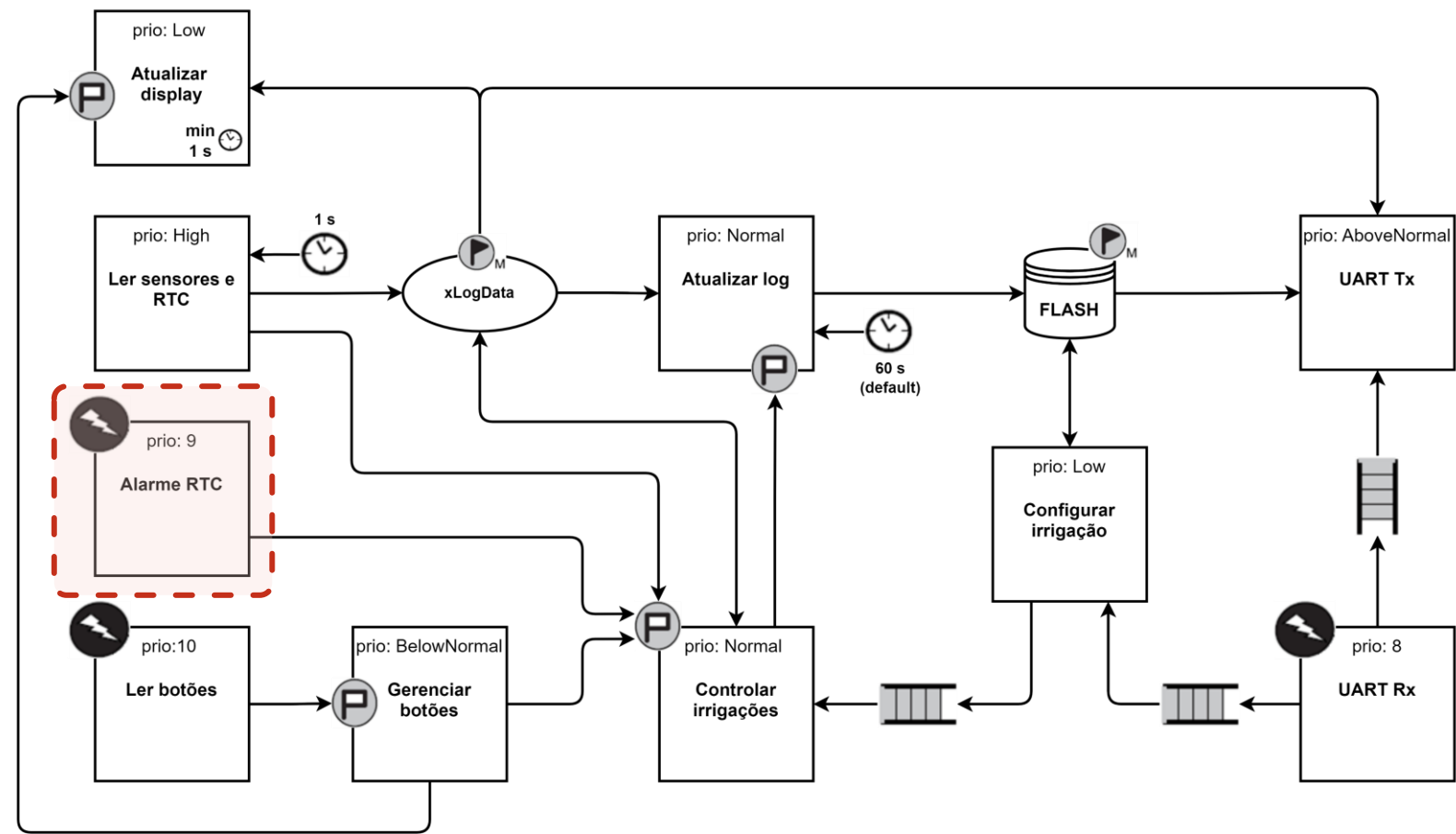
Nome	configureWatering
Prioridade	Low
Stack	500 Bytes
Detalhes	Configura regras de irrigação a partir de última configuração salva na FLASH e de comandos recebidos via fila de mensagens da ISR <i>receiveUART</i> . A cada nova configuração recebida pela UART, atualiza região da FLASH correspondente via mutex e envia mensagem por fila para tarefa <i>controlWatering</i> para atualizar limites de umidade e alarmes do RTC.

Componentes RTOS | *Tarefas*



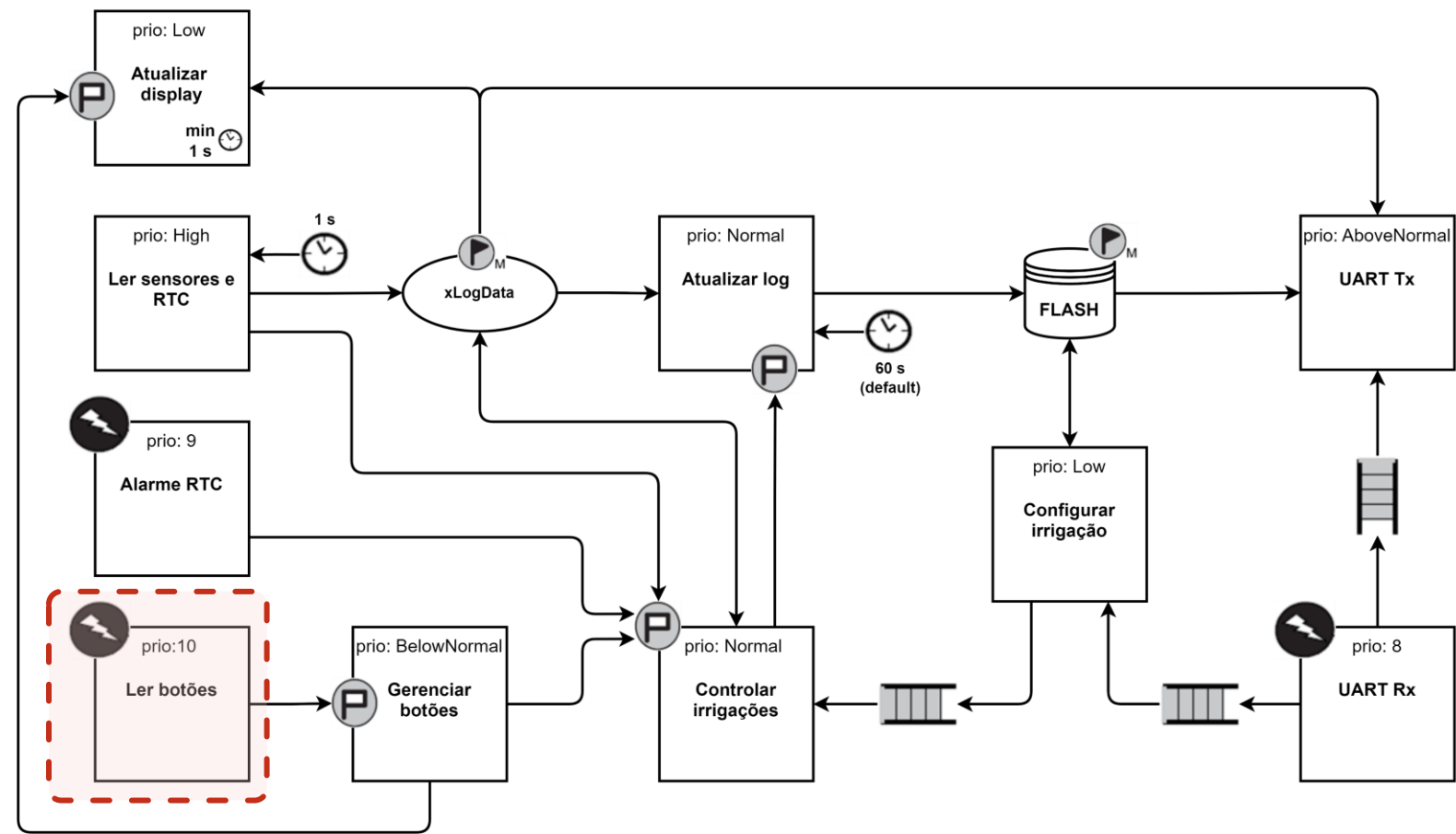
Tarefa: <i>UART Tx</i>	
Nome	transmitUART
Prioridade	AboveNormal
Stack	1 KB
Detalhes	Envia informações solicitadas via UART, como medidas dos sensores, status e configurações das irrigações e leitura do log. Recebe comandos da ISR <i>receiveUART</i> via fila de comandos e acessa informações da variável global <i>xLogData</i> e da FLASH via mutex.

Componentes RTOS | Interrupções (ISR)



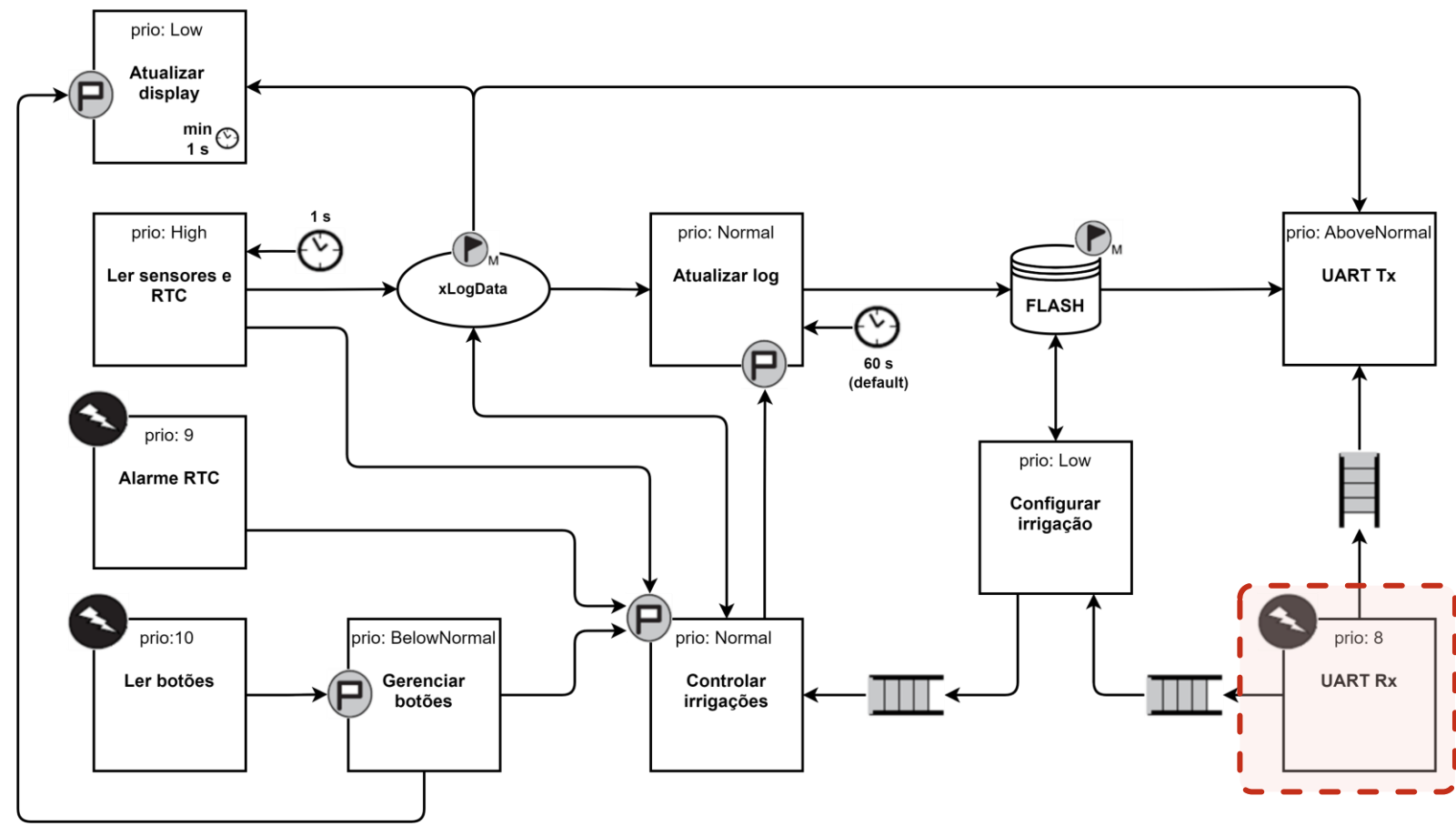
ISR: Alarme RTC	
Nome	alarmRTC
Prioridade	9
Disparada por	Alarmes A e B do RTC
Detalhes	Alarmes A e B do RTC correspondem, respectivamente, ao próximo acionamento e desacionamento de irrigações periódicas programadas. Os alarmes sinalizam a tarefa <i>controlWatering</i> via <i>thread flags</i> , que executa a lógica de acionamento e desacionamento das irrigações.

Componentes RTOS | Interrupções (ISR)



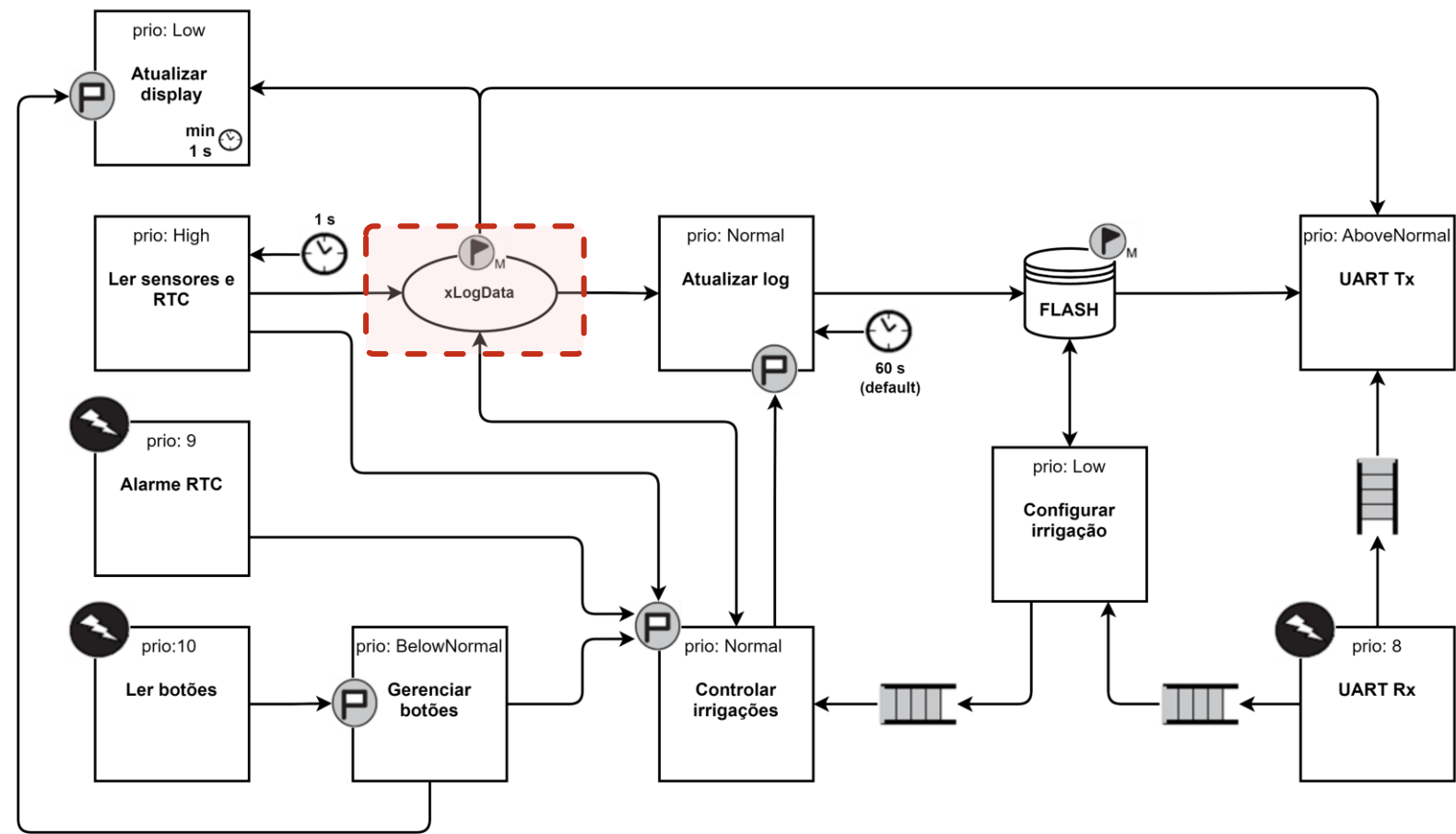
ISR: Ler botões	
Nome	readButtons
Prioridade	10
Disparada por	Módulo EXTI
Detalhes	Informa a tarefa <i>manageButtons</i> via <i>thread flags</i> qual dos dois botões foi acionado: seleciona canal (<i>selectChannel</i>) ou alterna estado do canal (<i>toggleChannelState</i>). Realiza <i>debouncing</i> , se necessário

Componentes RTOS | Interrupções (ISR)



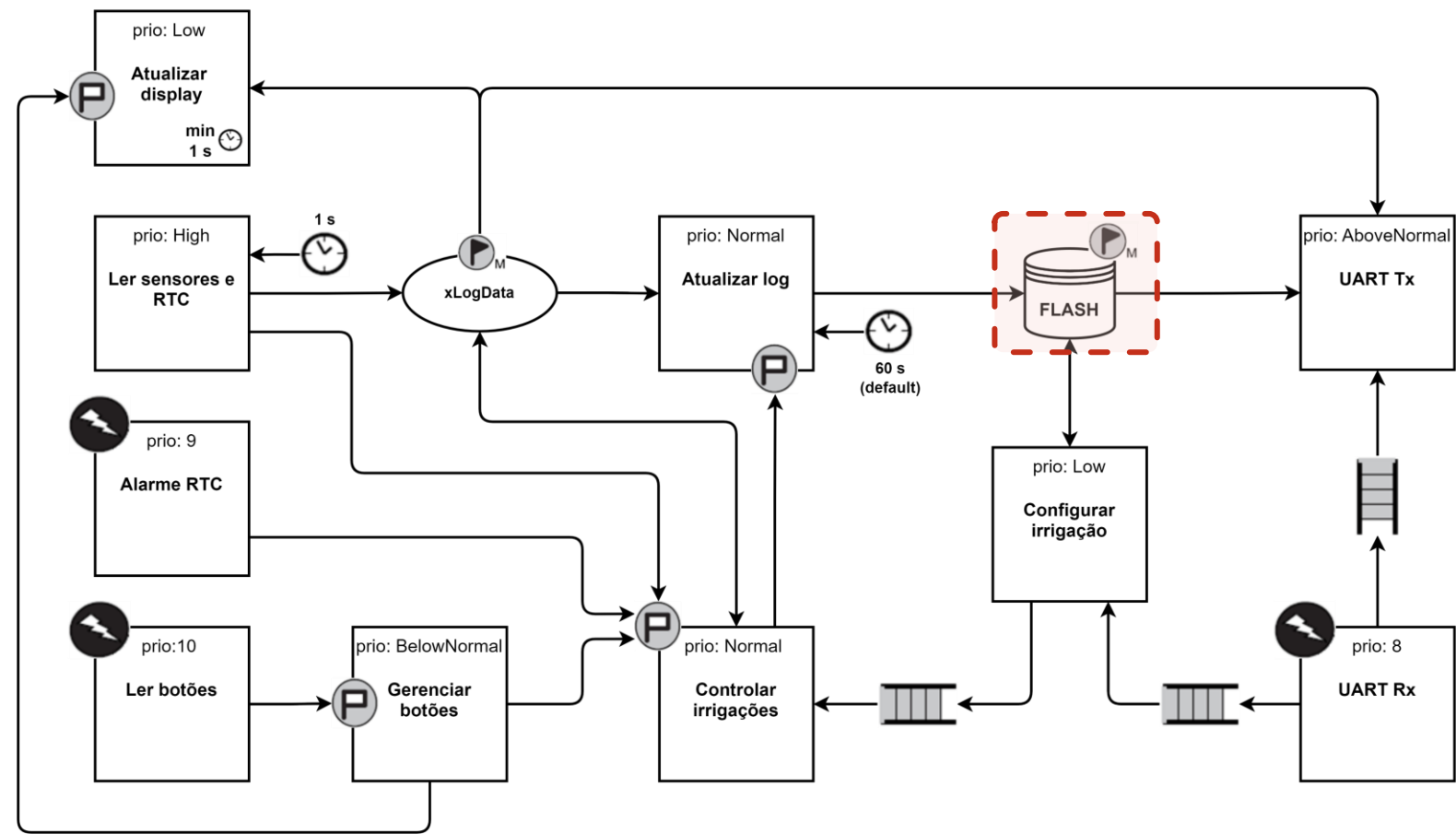
ISR: UART Rx	
Nome	receiveUART
Prioridade	8
Disparada por	Módulo UART
Detalhes	Recebe requisições via UART para gerenciar irrigações, configurar taxa de aquisição do log e solicitar leitura de medidas dos sensores e conteúdo do log. Gerenciamento de irrigações é realizado enviando mensagens através de fila para tarefa <i>configureWatering</i> . Leitura das medidas e do log sinalizada à tarefa <i>transmistUART</i> via <i>thread flag</i> .

Componentes RTOS | *Mutexes*



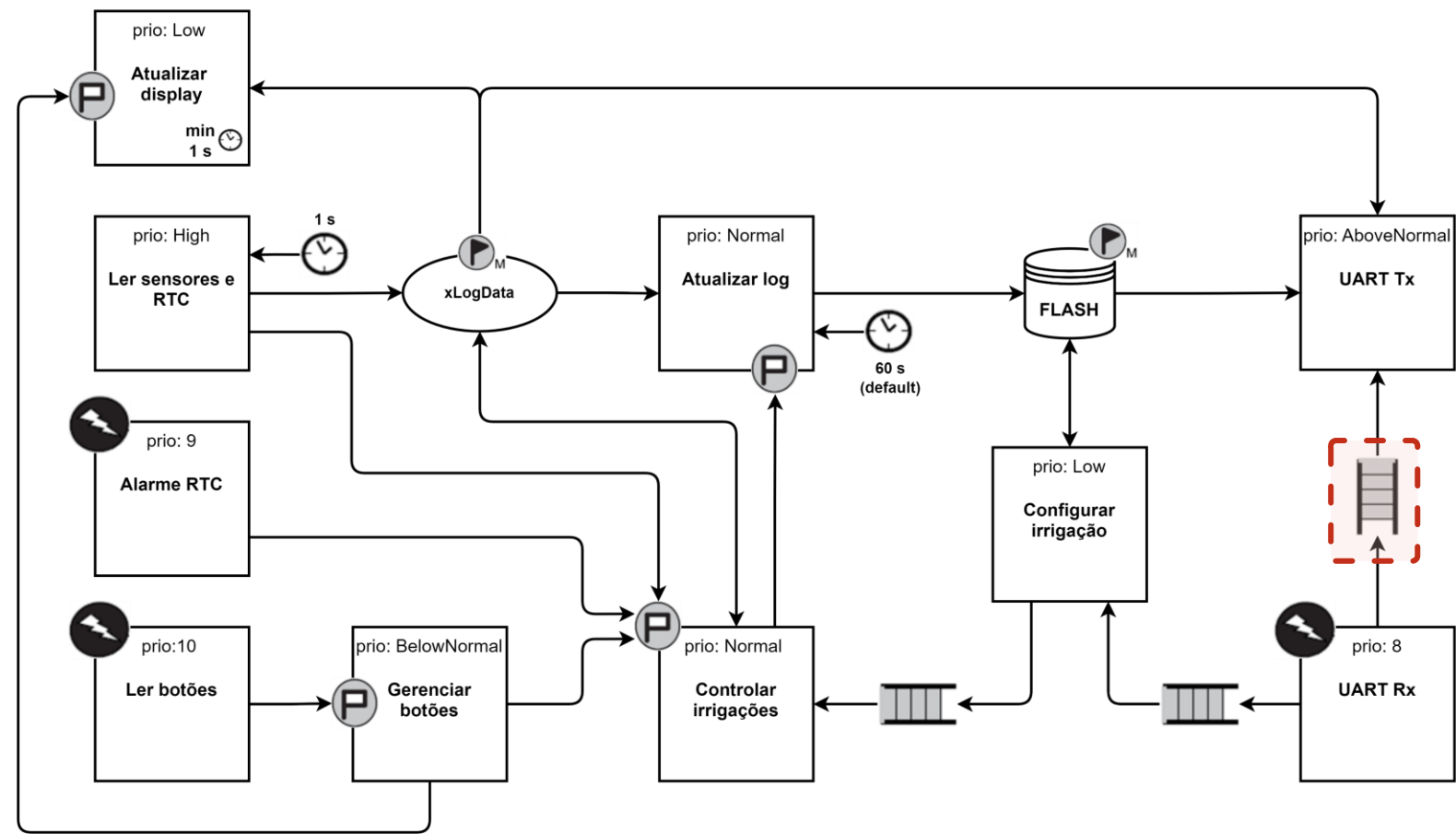
Mutex: Acesso à xLogData	
Nome	mutexLogData
Recurso protegido	xLogData (variável global)
Detalhes	Mutex para escrita e leitura da variável global <i>xLogData</i> , que contém data, hora, medidas e status dos canais de irrigação. Proteção do recurso garante atualização e leitura dos dados de forma consistente.

Componentes RTOS | *Mutexes*



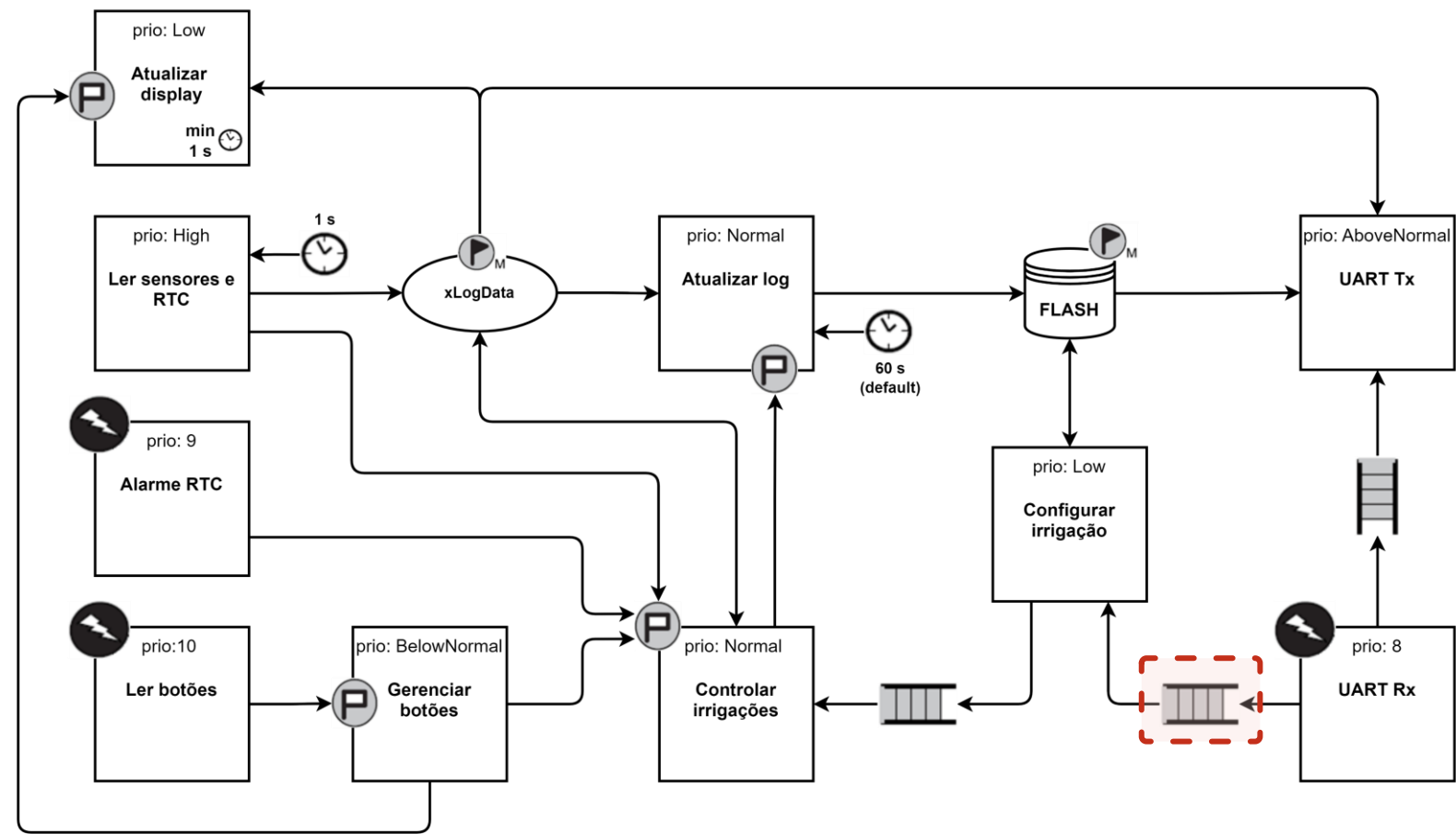
Mutex: Acesso à FLASH	
Nome	mutexFLASH
Recurso protegido	Memória FLASH de dados
Detalhes	Mutex para escrita e leitura da memória FLASH de dados contendo a configuração das regras de irrigação e o log de aquisição.

Componentes RTOS | *Queues*



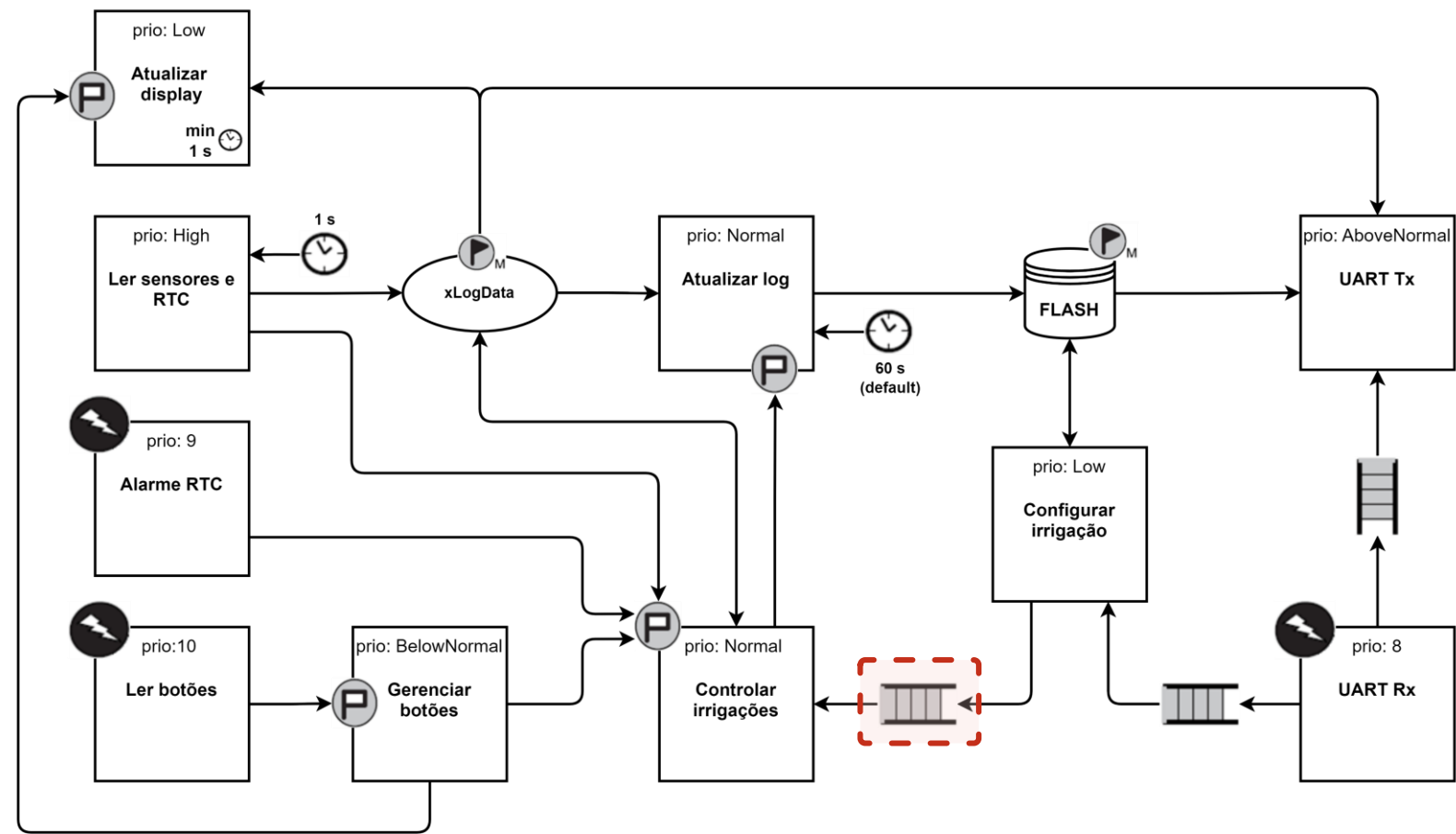
Queue: Responder UART	
Nome	queueReturnUART
Tamanho	16 x 1 byte
Dado	Comando a ser respondido via UART

Componentes RTOS | *Queues*



Queue: comando de configuração de irrigação	
Nome	queueConfigureWateringCommand
Tamanho	5 x 5 bytes
Dado	Comando para configurar irrigação <ul style="list-style-type: none">• 1 byte: canal + habilitar/desabilitar + tipo de regra (periódica/por umidade)• 2 bytes: período (em minutos) ou umidade mínima para acionar• 2 bytes: tempo de rega (em minutos) ou umidade máxima para desacionar

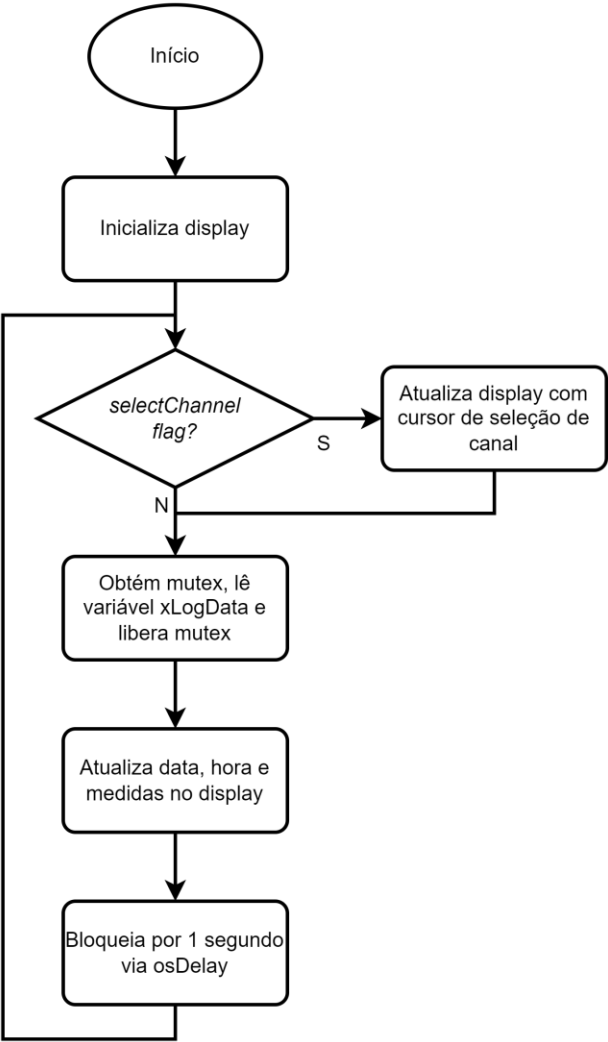
Componentes RTOS | *Queues*



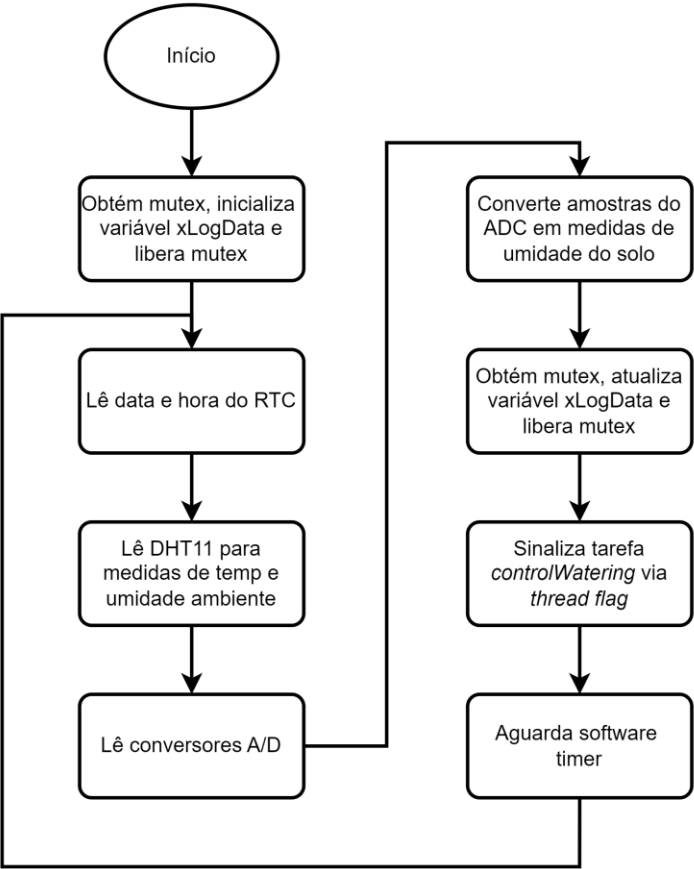
Queue: configuração de irrigação	
Nome	queueConfigureWateringControl
Tamanho	5 x 5 bytes
Dado	Mensagem para configurar controlador de irrigação a partir de UART ou configuração lida na FLASH durante inicialização <ul style="list-style-type: none">• 1 byte: canal + habilitar/desabilitar + tipo de regra (periódica/por umidade)• 2 bytes: período (em minutos) ou umidade mínima para acionar• 2 bytes: tempo de rega (em minutos) ou umidade máxima para desacionar

Fluxogramas das tarefas

updateDisplay

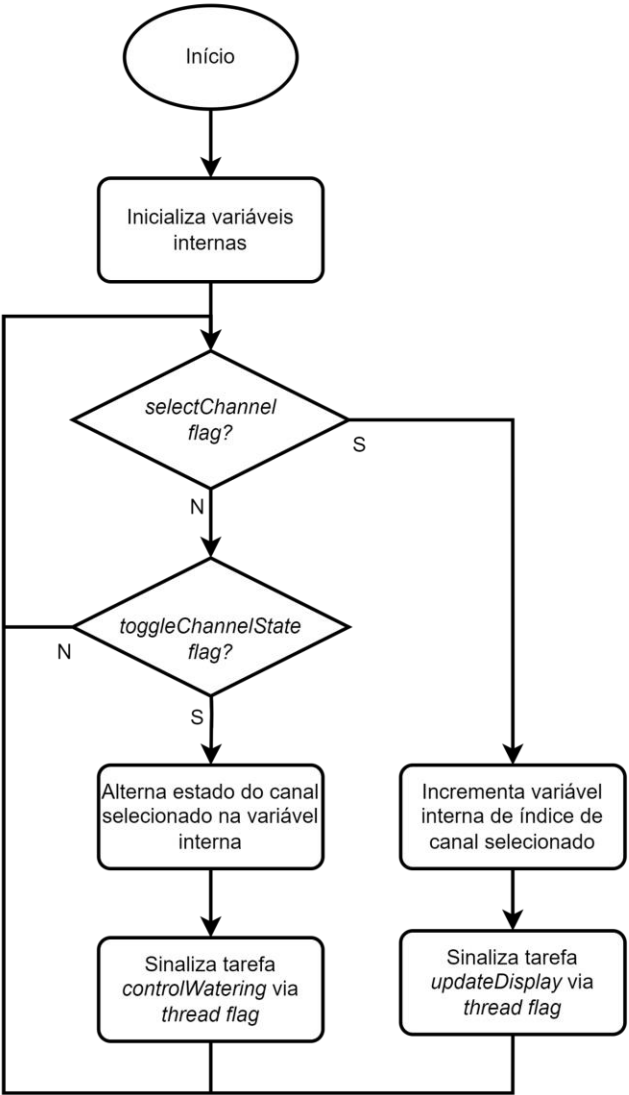


readSensorsRTC

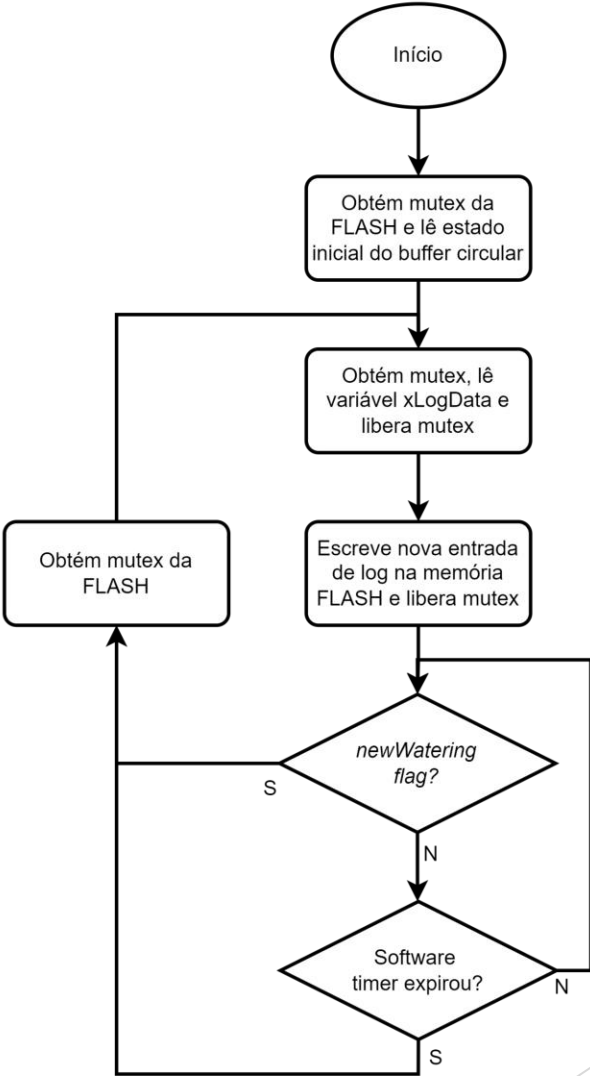


Fluxogramas das tarefas

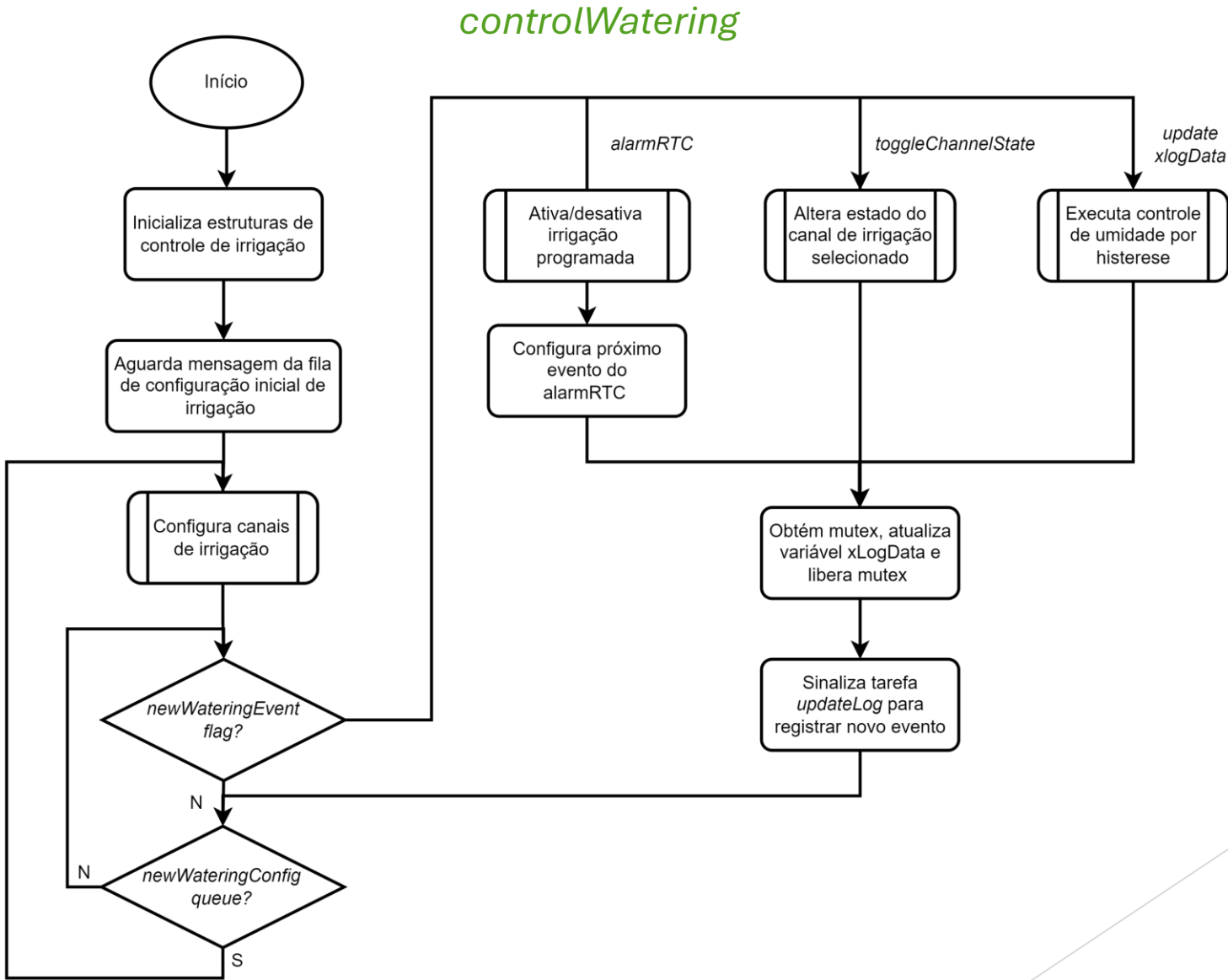
manageButtons



updateLog

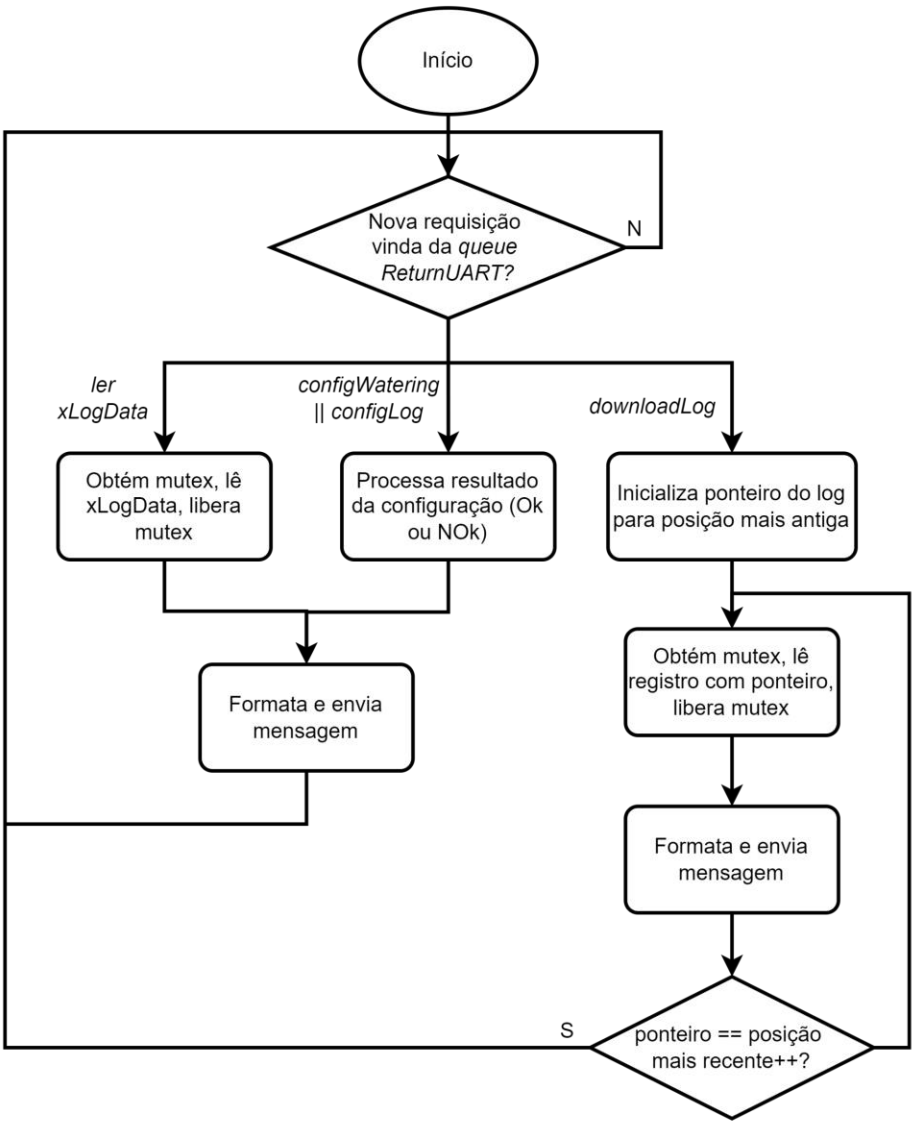


Fluxogramas das tarefas

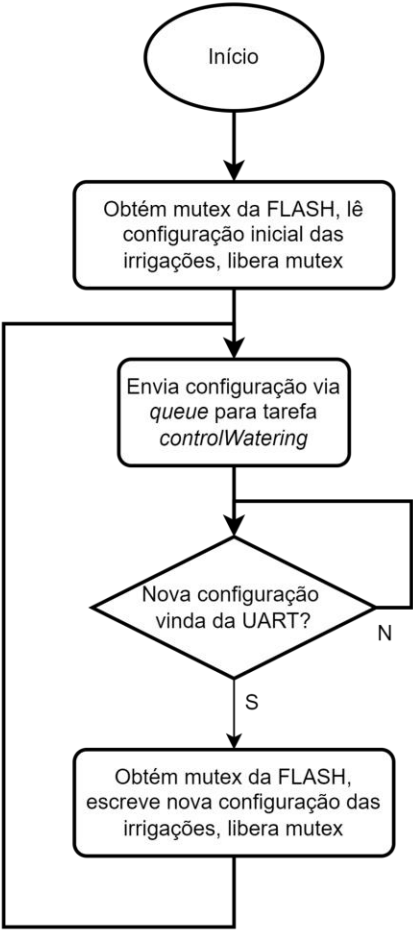


Fluxogramas das tarefas

transmitUART

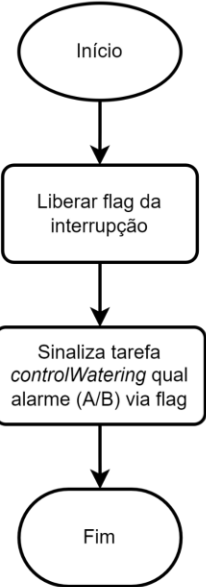


configureWatering

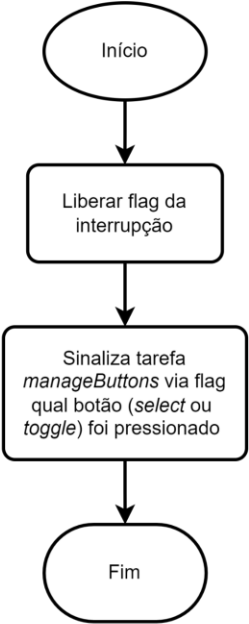


Fluxogramas das ISR's

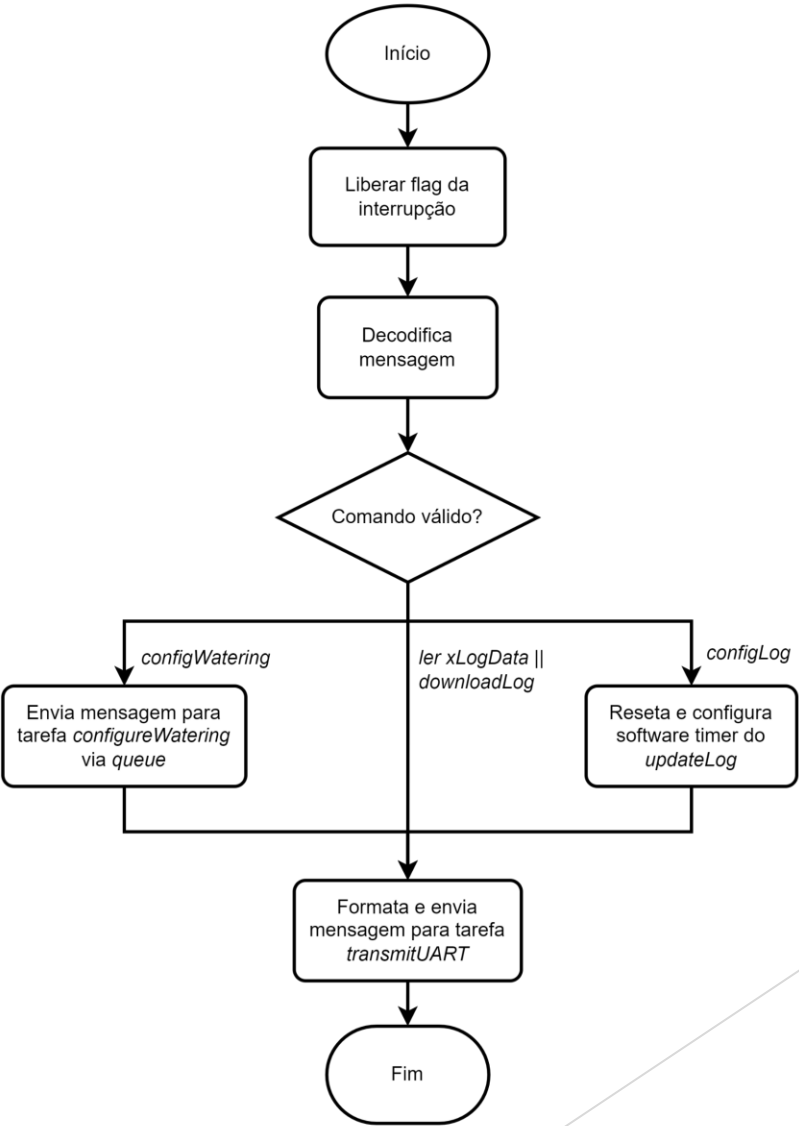
alarmRTC



readButtons



receiveUART



Formato do log de aquisição

- ▶ Buffer circular de 100 KB na memória FLASH, com ponteiros gerenciados pela tarefa *updateLog*
- ▶ Entradas no log como typedef struct contendo informações da tabela ao lado
- ▶ Tamanho total da struct vai depender se compilador vai otimizar (19 bytes) ou alinhar memórias pelo maior tipo (28 bytes)
- ▶ Para uma aquisição configurada a cada 60 segundos, a duração total do log será:
 - ▶ Caso otimizado: 89,8 horas (3,7 dias)
 - ▶ Caso alinhado: 60,95 horas (2,5 dias)

Campo	Tamanho [bytes]	Formato
day	1	uint8
month	1	uint8
year	1	uint8
hour	1	uint8
min	1	uint8
sec	1	uint8
ambientTemp	1	uint8
ambientHum	1	uint8
soilHum1	2	uint16
soilHum2	2	uint16
soilHum3	2	uint16
soilHum4	2	uint16
soilHum5	2	uint16
relaysStatus	1	uint8
Total otimizado [bytes]	19	
Tamanho total "alinhado" [bytes]	28	

Cronograma atualizado



Atividades	Semana / último dia							
	1	2	3	4	5	6	7	8
	08/out	15/out	22/out	29/out	05/nov	12/nov	19/nov	26/nov
Finalizar projeto de hardware	Planejado							
	Executado	Executado						
Compra e empréstimo de componentes		Planejado						
		Executado						
Detalhamento do projeto de software: definição do uso de periféricos e componentes do RTOS, especificação da memória de registro de dados		Planejado						
		Executado	Executado	Executado				
Programação e teste de periféricos			Planejado					
			Atrasado	Atrasado				
Especificação da aplicação em Python			Planejado					
			Iniciado	Iniciado				
Programação e teste de biblioteca de registro e de comunicação				Planejado				
				Atrasado				
Configuração dos componentes do RTOS				Planejado				
				Atrasado				
Integração das bibliotecas na aplicação RTOS					Planejado			
Implementação da aplicação em Python						Planejado		
Testes, validações e depuração							Planejado	
Finalização da escrita de relatório								Planejado