Embedded Thermometer

Germano Sobroza
Sat Aug 15 2020

Embedded Thermometer

Author

Germano Sobroza

1. This project works with the use of an Arduino Uno board, temperature sensor with a serial module and a push-button.
2. The temperature sensor is read in real time and the measurements are sent to the terminal and memory tasks.
3. If the push-button is pressed, an interruption is triggered and the averaged of the measures stored in memory are calculated and sent to the terminal task.

File Index

File List

Here is a list of all files with brief descriptions:

File Documentation

main.cpp File Reference

```
#include <Arduino.h>
#include <Arduino_FreeRTOS.h>
#include <DallasTemperature.h>
#include <OneWire.h>
#include <EEPROM.h>
#include <queue.h>
#include <stdio.h>
```

Macros

- #define **DS18B20** 7
  Define the digital pin used by the sensor.

- #define **MEMORY_SIZE** 1024/sizeof(float)
  Memory lenght.

Functions

- void **sensorTask** (void *pvParameters)
- void **terminalTask** (void *pvParameters)
- void **eepromTask** (void *pvParameters)
- void **buttonInterrupt** ()
  buttonInterrupt() *deals with an interrupt generated by an external button. The ISR calls CalMedia function and send the mean the temperature throught queue_2*

- float **CalcMedia** ()
  CalMedia() is called by the ***buttonInterrupt()*** *and calculates the mean of all temperatures wroten on the EEPROM.*

- void **clearEEPROM** ()
  clearEEPROM() *set all EEPROM bits to 0*

- char * **ftoa** (char *a, double f, int precision)
- OneWire **ourWire** (**DS18B20**)
- void **setup** ()
  The ***setup()*** *function intialize the serial monitor, the DallasTemperature sensor, configure the interrupt pin, clears the EEPROM memory, creates two queues and three tasks.*

- void **loop** ()
- void **sensorTask** (void *pvParameters __attribute__((unused)))
  sensorTask() *reads the temperature from the sensor, convert it to a String type and send it througth queue_1 to* ***terminalTask()****. After that, queue_2 sends the float value of the temperature to eepromTask.*

- void **terminalTask** (void *pvParameters __attribute__((unused)))
  terminalTask() *waits for messages sent thought queue_2 and print it in the terminal.*

- void **eepromTask** (void *pvParameters __attribute__((unused)))
  eepromTask() *reiceves float temperature and writes in the EEPROM, addressed by memory_index. If memory_index is bigger than MEMORY_SIZE, memory_indez is set to -sizeof(float), working like a circular buffer.*

Variables

- const byte **interruptPin** = 2
  Interrup pin used by the button.

- int **memory_index** = -1*sizeof(float)
  *Memory index variable, initialized with -1*sizeof(float) for implentation purposes.*

- bool **FULL_MEMORY** = false
  when all positions have been written with some temperature, turn into true

- DallasTemperature sensors & **ourWire**
- QueueHandle_t **queue_1**
- QueueHandle_t **queue_2**

Macro Definition Documentation

#define DS18B20 7

Define the digital pin used by the sensor.

Definition at line 28 of file main.cpp.

#define MEMORY_SIZE 1024/sizeof(float)

Memory lenght.

Definition at line 29 of file main.cpp.

Function Documentation

void buttonInterrupt ()

buttonInterrupt() deals with an interrupt generated by an external button. The ISR calls CalMedia function and send the mean the temperature throught queue_2

Definition at line 203 of file main.cpp.

float CalcMedia ()

CalMedia() is called by the **buttonInterrupt()** and calculates the mean of all temperatures wroten on the EEPROM.

Definition at line 173 of file main.cpp.

void clearEEPROM ()

clearEEPROM() set all EEPROM bits to 0

Definition at line 225 of file main.cpp.

void eepromTask (void *pvParameters __*attribute*__(unused))

eepromTask() reiceves float temperature and writes in the EEPROM, addressed by memory_index. If memory_index is bigger than MEMORY_SIZE, memory_indez is set to -sizeof(float), working like a circular buffer.

Definition at line 150 of file main.cpp.

void eepromTask (void * *pvParameters*)

char* ftoa (char * *a*, double *f*, int *precision*)

void loop ()

Definition at line 97 of file main.cpp.

OneWire ourWire (DS18B20)

void sensorTask (void *pvParameters __*attribute*__(unused))

sensorTask() reads the temperature from the sensor, convert it to a String type and send it througth queue_1 to **terminalTask()**. After that, queue_2 sends the float value of the temperature to eepromTask.

< stores the read temperature

Definition at line 109 of file main.cpp.

void sensorTask (void * *pvParameters*)

void setup ()

The **setup()** function intialize the serial monitor, the DallasTemperature sensor, configure the interrupt pin, clears the EEPROM memory, creates two queues and three tasks.

Definition at line 44 of file main.cpp.

void terminalTask (void *pvParameters __*attribute*__(unused))

terminalTask() waits for messages sent thought queue_2 and print it in the terminal.

Definition at line 130 of file main.cpp.

void terminalTask (void * *pvParameters*)

Variable Documentation

bool FULL_MEMORY = false

when all positions have been written with some temperature, turn into true
Definition at line 32 of file main.cpp.

const byte interruptPin = 2

Interrup pin used by the button.
Definition at line 30 of file main.cpp.

int memory_index = -1*sizeof(float)

Memory index variable, initialized with -1*sizeof(float) for implentation purposes.
Definition at line 31 of file main.cpp.

DallasTemperature sensors& ourWire

Definition at line 35 of file main.cpp.

QueueHandle_t queue_1

Definition at line 36 of file main.cpp.

QueueHandle_t queue_2

Definition at line 37 of file