Team 27

Embedded Project

Heater Feedback

Students:

Ahmed Amr Mohamed	18P1645
Adel Asaad Sarofeim	18P2949
Marwan Atef Hamed	18P8678
Mohamed Adel Lotfy	18P1724
Omar Khaled Shawky	14P6071

Supervisor:

Dr. Sherif Hammad

Eng. Ahmed Farouq

Table of Contents

Application: 2

Flowchart: 3

Wiring Sim: 3

Files Description: 4

- Main 4
- Dio 4
- Dio-Config 4
- Adc 4
- Lcd 4
- Macro.h 5
- Types.h 5
- Platform_types.h 5
- tm4c123gh6pm.h 5

Drive Link: 5

Application:

This application is going to simulate a working feedback heater that is supposed to keep the temperature above a certain level.

This application uses semaphores to prevent racing conditions, uses interrupts for its tasks to display, and check temperature, to turn off/on heater, and to reset the wanted temperature.

This application starts by taking the wanted temperature from user input using a keyboard and its UART system, the system then checks for the current temperature (simulated by a potentiometer) and it takes an analog input, then converts it into a digital value for processing.

The current temperature measured is then compared with the wanted temperature input by the user, if the measured temperature is lower than the wanted temperature the heater is then turned on, and vice versa if the measured temperature is higher than the wanted temperature the heater is then turned off.

The heater is simulated by a GREEN LED.

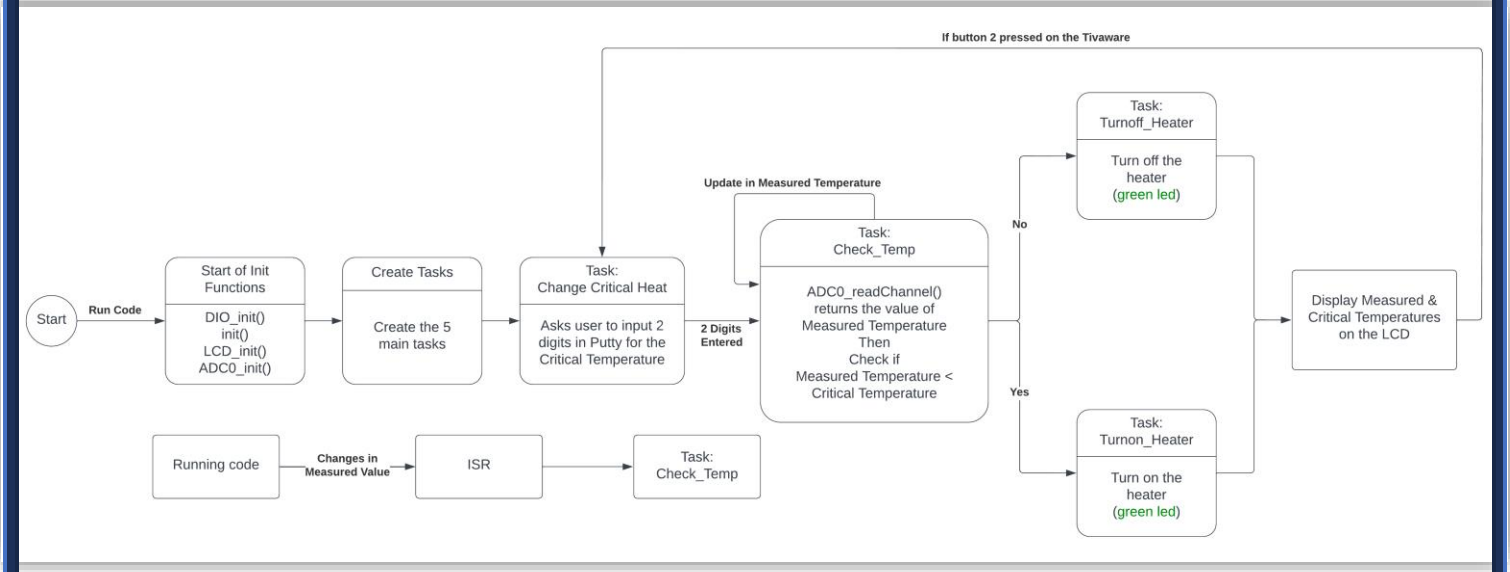
The user can reset the wanted temperature by pressing on a button on the TivaC processor (button 2) which then prompts the user to re-input their wanted temperature using the keyboard again.

While the device is waiting for a wanted temperature (input from the user) the device is otherwise inaccessible in all other ways as the flag that is raised by pressing button 2 cannot be lowered until a value is received.

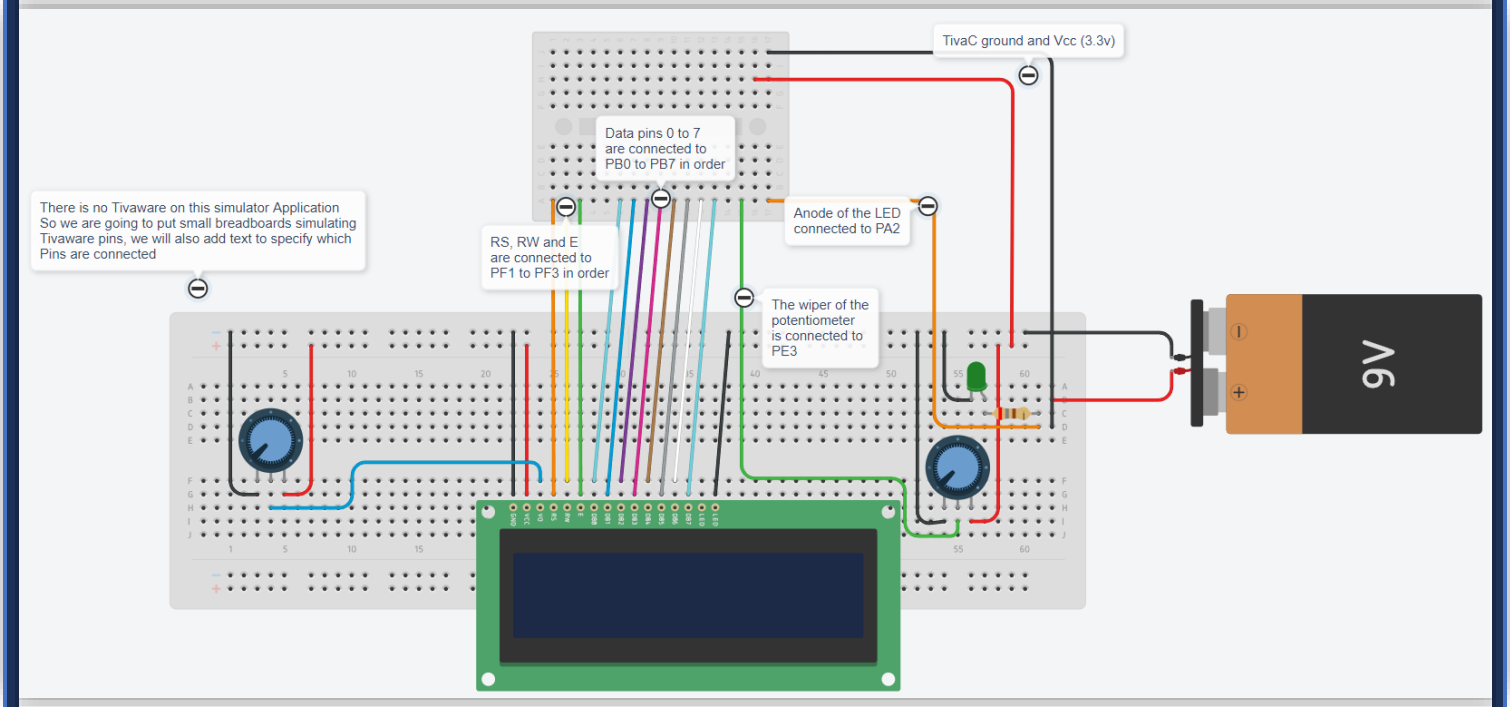
The device (if uninterrupted by other tasks) keeps updating the LCD to show current temperature measured (from potentiometer) and the wanted temperature (input from user).

Current temperature measured is accurate to 0.5

Flowchart:



Wiring Sim:



Files Description:

This part will contain a description for each combination of a c file and its own header, alongside any configuration files when needed.

Main

This is the main file application, where we put our driver code, it consists of initialization code to initialize and setup the TivaC pins, clocks, timers, interrupts, semaphores, and memory.

Then it contains logic for how each task operates and whether or not its periodic.

Basically, the main brain of the application.

Dio

The Dio file contains the functions needed to read/write pins and whole ports when needed from the TivaC, it uses its own configuration file which will be mentioned next.

Dio-Config

The configuration file contains the custom data structure for a pin in TivaC

Adc

The adc file is the logic that handles reading analog input from a channel input from TivaC and turning it into a digital value.

Lcd

The lcd file has a lot of macros defined for ease of accessibility to data on the LCD that we're working on in the project.

Alongside the macros it also comes with its own API functions to automatically initialize the LCD, clear it, write on it, and change where the cursor is.

Macro.h

The Macro.h file is a stand-alone header file containing some macros to set, toggle, clear, read, and check bits in the memory.

Types.h

The Types.h file is a stand-alone header file containing typedefs for custom used types in the project.

Platform_types.h

The platform_types.h file is a stand-alone header file containing macros for TivaC port values, pin numbers, pin values for each port.

tm4c123gh6pm.h

This is the attached macro definition file supplied by tivaware.

Drive Link:

https://drive.google.com/drive/folders/1mnV-dVydNLyaKR-cMtSHaxjhgWYUorLo?usp=sharing