

**Smart Home with FreeRTOS**

**By: Group 8**

**Abstract:**

Smart Home consists of 3 subsystems using two Tiva C as shown in figure.

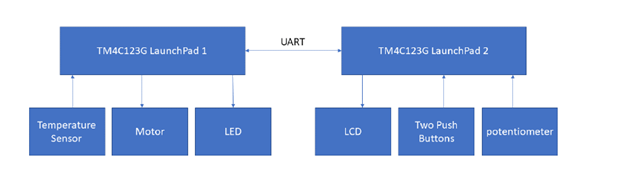
**Components used:**

1-Two TM4C123G Launchpads

2-LCD

3-Stepper Motor

4-Potentiometer



**Port Drivers:**

|  |  |
| --- | --- |
| Name | Port\_Init |
| Input | uint8\_t port\_index |
| Return | Void |
| Description | Initialize port based on port index |

|  |  |
| --- | --- |
| Name | Port\_SetPinDirection |
| Input | uint8\_t port\_index,  uint8\_t pins\_mask, Port\_PinDirectionType pins\_direction |
| Return | Void |
| Description | Change the direction of the selected pins |

|  |  |
| --- | --- |
| Name | Port\_SetPinPullUp |
| Input | uint8\_t port\_index,  uint8\_t pins\_mask,  uint8\_t enable |
| Return | Void |
| Description | Make the Pins chosen pulled up internally with resistance |

|  |  |
| --- | --- |
| Name | Port\_SetPinPullDown |
| Input | uint8\_t port\_index, uint8\_t pins\_mask, uint8\_t enable |
| Return | Void |
| Description | Make the Pins chosen pulled down internally with resistance. |

**ADC Drivers:**

|  |  |
| --- | --- |
| Name | ADC0\_init |
| Input | uint8\_t channel\_num,  triggerType trig\_type |
| Return | Void |
| Description | Initialization for ADC |

|  |  |
| --- | --- |
| Name | ADC0\_channelConfig |
| Input | uint8\_t channel\_num |
| Return | Void |
| Description | Configure chosen ADC channel to work as analog input |

|  |  |
| --- | --- |
| Name | ADC0\_start\_conversion |
| Input | Void |
| Return | Void |
| Description | Trigger ADC to start conversion |

|  |  |
| --- | --- |
| Name | ADC0\_digital\_toVoltage |
| Input | uint16\_t digital\_value |
| Return | Float |
| Description | Converts pulses (output from ADC) to voltage |

**Temperature Sensor Driver:**

|  |  |
| --- | --- |
| Name | TempSensor\_init |
| Input | Void |
| Return | Void |
| Description | Initialize temperature sensor |

|  |  |
| --- | --- |
| Name | TempSensor\_readTemp |
| Input | Void |
| Return | Float |
| Description | Return temperature read from sensor |

**Potentiometer Driver:**

|  |  |
| --- | --- |
| Name | Read\_Potentiometer |
| Input | Void |
| Return | Float |
| Description | Return potentiometer value |

**Interrupt Driver:**

|  |  |
| --- | --- |
| Name | Initalize\_PORTF\_interrupt |
| Input | uint8\_t pinsmask |
| Return | Void |
| Description | Initialize interrupt on the pins of ports F selected by pinsmask |

**PWM Driver**:

|  |  |
| --- | --- |
| Name | PWM\_Init |
| Input | Void |
| Return | void |
| Description | |  | | --- | |  |   Initialize the PWM timer |

|  |  |
| --- | --- |
| Name | PWM\_set\_dutyCycle |
| Input | Uint16\_t duty\_cycle |
| Return | void |
| Description | |  | | --- | |  |   Sets the PWM duty cycle |

**LCD Driver:**

|  |  |
| --- | --- |
| Name | LCD\_SendCommand |
| Input | Uint8\_t com |
| Return | Void |
| Description | Sends command to LCD.  EX: clear, display, shift, etc.… |

|  |  |
| --- | --- |
| Name | LCD\_SendData |
| Input | Uint8\_t data |
| Return | Void |
| Description | |  | | --- | |  |   Sends data to LCD |

|  |  |
| --- | --- |
| Name | LCD\_ShowString |
| Input | Const char \*str |
| Return | Void |
| Description | Takes a long string and print it on the LCD |

|  |  |
| --- | --- |
| Name | LCD\_init |
| Input | void |
| Return | Void |
| Description | Make initialization of LCD |

|  |  |
| --- | --- |
| Name | LCD\_ClearScreen |
| Input | void |
| Return | Void |
| Description | |  | | --- | |  |   Use the function command to Clear screen |

|  |  |
| --- | --- |
| Name | LCD\_GoToRowColumn |
| Input | Uint8\_t row  Uint8\_t col |
| Return | Void |
| Description | |  | | --- | |  |   Put the cursor of printing on specific position |

|  |  |
| --- | --- |
| Name | LCD\_ShowStringRowAndColumn |
| Input | Uint8\_t row  Uint8\_t col  Const char \*str |
| Return | Void |
| Description | Print string from LCD\_GoToRowColumn position |

|  |  |
| --- | --- |
| Name | Itoa |
| Input | Int I  Char b[] |
| Return | Char\* |
| Description | |  | | --- | |  |   Convert int to char |

|  |  |
| --- | --- |
| Name | LCD\_IntgerToString |
| Input | Uint32\_t Data |
| Return | Void |
| Description | |  | | --- | |  |   Print int number on the screen |

**DIO Driver:**

|  |  |
| --- | --- |
| Name | DIO\_ReadPort |
| Input | Uint8\_t port\_index  Uint8\_t pins\_mask |
| Return | Uint8\_t |
| Description | |  | | --- | |  |   Return the value Chosen by the port\_index |

|  |  |
| --- | --- |
| Name | DIO\_WritePort |
| Input | Uint8\_t port\_index  Uint8\_t pins\_mask  Dio\_LevelType pins\_level |
| Return | Uint8\_t |
| Description | Return the value Chosen by the port\_index |

|  |  |
| --- | --- |
| Name | DIO\_FlipPort |
| Input | Uint8\_t port\_index  Uint8\_t pins\_mask |
| Return | Void |
| Description | Toggle the value of the pin selected |

**UART Driver:**

|  |  |
| --- | --- |
| Name | UART7\_init |
| Input | Bool intEn  Uint32\_t baud rate |
| Return | Void |
| Description | Initialization for UART 7 |

|  |  |
| --- | --- |
| Name | UART7\_Send\_NonBusy |
| Input | Init8\_t data |
| Return | bool |
| Description | Sends data to UART without busy waiting and returns 1:if data was successfully was send and 0 otherwise |

|  |  |
| --- | --- |
| Name | UART7\_send\_data\_packet |
| Input | Char title[]  Init8\_t data |
| Return | Void |
| Description | |  | | --- | |  |   Sends packet to UART consisting of two words: the first letter of the title and data |

|  |  |
| --- | --- |
| Name | UART7\_Recieve\_Busy |
| Input | void |
| Return | Init8\_t |
| Description | Receive data from UART with busy waiting |

|  |  |
| --- | --- |
| Name | UART7\_Recieve\_NonBusy |
| Input | Init8\_t \*data |
| Return | Bool |
| Description | Receive data from UART without busy waiting and returns 1 : If data was successfully received and 0 otherwise |

|  |  |
| --- | --- |
| Name | UART7\_recieve\_packet |
| Input | void |
| Return | Void |
| Description | |  | | --- | |  |   Receive packet from UART consisting of two words: the first letter of the title and data , and dumping the data in one of the defined global variables depending on the title |

|  |  |
| --- | --- |
| Name | UART7\_Handler |
| Input | void |
| Return | Void |
| Description | |  | | --- | |  |   The handler is called when the receive fifo reaches a threshold of 4 words OR if it didn’t reach the threshold for a long time .  -On threshold it takes the packets and dumps data in global variables  -On receive timeout it flushes the fifo from the uncomplete packet |

**Functions:**

|  |  |
| --- | --- |
| Name | xTaskCreate |
| inputs | Task Function  Parameters  Stack size  Priority |
| outputs | Nothing |
| Description | Makes a space for a task in the FreeRTOS working area. |

|  |  |
| --- | --- |
| Name | vTaskStartScheduler |
| inputs | Nothing |
| outputs | Nothing |
| Description | Start the Scheduling process between tasks (SHOULD NOT RETURN). |

|  |  |
| --- | --- |
| Name | vTaskDelay |
| inputs | Required delay in ticks |
| outputs | Nothing |
| Description | Moves the task to the blocking state until the specified delay is finished and moves it back to the ready state. |

|  |  |
| --- | --- |
| Name | taskYIELD |
| inputs | Nothing |
| outputs | Nothing |
| Description | Moves the calling task to the ready queue and calling the next one to the running queue. |

We used the idea of Mailboxes to synchronize data usage between tasks.

A screenshot of a cell phone

Description automatically generated

Tiva 1:

|  |  |
| --- | --- |
| Task\_Name | vTempTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic Task every 1000ms to get the ADC value of the Temp sensor. |

|  |  |
| --- | --- |
| Name | vUARTRecieveTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic task that checks if there was something in the UART buffer to write it in the PWM led and then transfer the CPU to the next task |

|  |  |
| --- | --- |
| Name | vUARTSendTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic task that tries to send checks if there was something in the UART buffer and then transfer the CPU to the next task |

Tiva 2:

|  |  |
| --- | --- |
| Task\_Name | vLCDTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic Task every 1000ms to show the value of Temperature received from UART. |

|  |  |
| --- | --- |
| Name | vUARTRecieveTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic task that checks if there was something in the UART buffer to write the Temperature data to its MailBox and then transfer the CPU to the next task. |

|  |  |
| --- | --- |
| Name | vUARTSendTask |
| inputs | Nothing |
| outputs | Nothing |
| Description | A periodic task that tries to send Potentiometer data and then transfer the CPU to the next task. |