

General Description

- This program is built to perform functions of a remote thermal sensor unit
- The major function of this remote sensor device is to collect the thermal temperature and current time stamp data to display on the LCD, then send it back to remote data storage server or remote console
- This program can be executed periodically by adding it as a “crontab” task. The command will be provided later.
- Future enhancement: Add motion detection by using interrupt function on GPIO

How to use this code

- **Use GNU make to execute the “makefile”. The “smart_sensor” ELF executable file “smart_sensor” will be generated accordingly**
- **At command prompt, type below 2 options to select the option to send thermal data:**

a. To remote server (e.g Server at 172.21.1.10)

```
>> ./smart_sensor 1 172.21.1.10
```

b. To remote console

```
>> ./smart_sensor 2
```

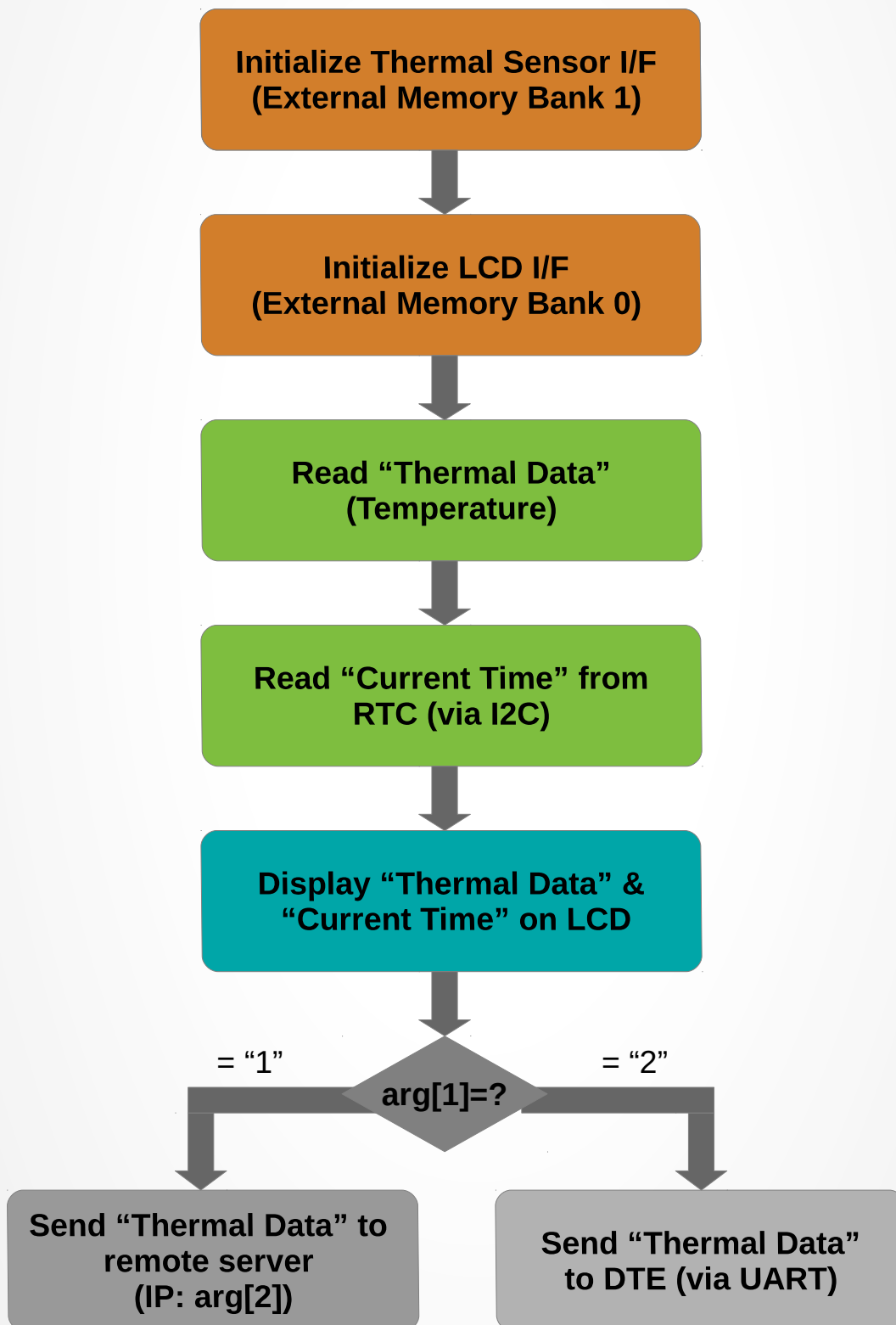
- **Add it as a periodical task:**

```
>> crontab -e
```

and add this line to run this program every 5 min

```
*/5 * * * * /path/smart-sensor 1 172.21.1.10
```

Flow Chart



Key Modules: Thermal_Read.c

- This module is used to read the thermal data from sensor LM35 via a 8-bit ADC convertor ADC0804. The I/F between ADC0804 and S3C4510 is external memory I/F. Below are the main functions performed by this modules.
- Initialize external memory bus bank 1 by configuring the bus width to 8-bit.
- Write “0” to ADC0804 to trigger the A-to-D conversion.
- Polling GPIO P0 to check if the conversion is finished.
- Read the 8-bit data from ADC0804.
- Convert the 8-bit data to “double” data value by applying voltage-to-degree conversion formula.
- Please refer to LM35, ADC0804 data-sheet and H/W Block description for more details.

Key Modules: LCD.c

- This module includes some main functions related to LCD display module HD44780. Below are the detailed descriptions of each functions.
- LCD_init():
 - a. Initialize external memory bus bank 1 by configuring the bus width to 8-bit
 - b. Configure basic parameters for LCD: Rows, Data Width, Display on-off, Cursor blinking mode..etc
- Clear_LCD(): Clear LCD screen.
- Write_LCD_Cmd(): Write command or RAM address into instruction register
- Wriet_LCD_Data(): Write data into data register.
- Check_LCD_Busy(): Check “Busy Flag”, has to be done before any read/write actions.
- GotoXY(): Move cursor to specific position.
- Note: Please refer to HD44780 data-sheet for more details about instruction & data register definitions.

Key Modules: I2C_RTC.c

- This module includes functions to read the timing data from RTC8564 via I2C I/F. Below are the details of key functions.
- I2C_init():
 - a. Setup I2C clock by configuring pre-scale value.
 - b. Enable interrupt and auto-ACK.
- I2C_RTC_Read: Perform below step for reading RTC data
 - a. Send out the slave address of RTC8564 to shift buffer register
 - b. Start the transmission.
 - c. Send the target register address to shift buffer register
 - d. Restart and set up Buffer Flag
 - e. Delay and read the data from shift buffer f. Stop this transaction.
- RTC_GET_TIME: Read the sec, min, hour, day, month data from RTC
- Note: Please refer to RTC8564 data-sheet for more details about timing register definitions and S3C4510 data-sheet for the steps to initialize I2C transactions.

Key Modules: UART.c

- This module includes functions to send out data to UART port. Below are the details of key functions.
- `open_port()`: Open the `"/dev/ttyS1"` port (COM port)
- `init_serial()`: Configure BAUD rate, data length
- `Read/write_data()`: read or Write data to UART port

Key Modules: Remote_Send.c

- This module is used to send out data to specific remote terminal. The IP is introduced by arg[2] in main function.
- In this program, IPv4 and TCP are used as network protocols.
- The remote terminal can be a data storage server to store the thermal data.