Meditech control panel software

1.0beta

Generated by Doxygen 1.8.6

Sun Jun 28 2015 14:10:51

Contents

Chapter 1

Main Page

This application will run on the ChipKit PI module of the Meditech device. The module is connected to the Raspberry PI Mod B2. The software includes two parts: one independent series of features, related to the Meditech general health status that can override the user controlled parameters. The board also control che alphanumeric LCD display for status messages, warnings and alarms.

The user-controlled part is managed by the Raspberry PI "PI master" that control the active probes and other functions. The counterpart of this application is a C++ command set running on the PI linux machine and communicates with the board through a simple serial protocol. To reduce the weight of this microcontroller application most of the informational strings shown by the control panel display are send by the linux machine.

Note

This code is distributed under the Apache license These sources has been developed under the mpide application, adapted for the ChipKit PI board. This application is part of the Meditech project, created by Enrico Miglino for Balearic Dynamics - Balearic Islands - Spain
For the last updated application version and subversion, see the version.h include file.

Author

Enrico Miglino enrico.miglino@gmail.com
Balearic Dynamics - Spain balearicdynamics@gmail.com

Version

1.0b

Date

First version on July 2015

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

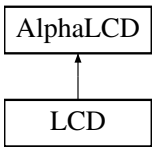
Class Documentation

5.1 LCD Class Reference

Manages the Alphanumeric display for program output messages.

`#include "LCD.h"`

Inheritance diagram for LCD:



Public Member Functions

- LCD ()
- ~LCD ()
- void enable (bool s)

  Set the display on or off.
- void blink (bool set)

  Set blink mode.
- void error (String m)

  shows an error message
- void error (String m, int x, int y)

  shows an error message at specified coordinates
- void message (String m)

  shows a string message
- void message (String m, int x, int y)

  shows a string message at specified coordinates
- void clean ()

  clean the LCD screen
- void dec (int n)

  shows an integer in decimal format
- void hex (int n)

  shows an integer in hexadeciaml format
- void bin (int n)

  shows an integer in binary format

- void oct (int n)

  shows an integer in octal format
- void welcome ()

  shows the program welcome message
- void menu (String sect1, String sect2, String sect3, String sect4)

Private Member Functions

- LCD (const LCD &c)
- LCD & operator= (const LCD &c)

Private Attributes

- AlphaLCD lcd

  AlphaLCD class inherited instance.

5.1.1 Detailed Description

Manages the Alphanumeric display for program output messages.

This class implements the *AlphaLCD* class that manages the Alphanumeric LCD display hardware using three digital pins via a shift-out register.

Definition at line 48 of file LCD.h.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 LCD::LCD ( )

5.1.2.2 LCD::~LCD ( )

5.1.2.3 LCD::LCD (const LCD & *c*) [private]

5.1.3 Member Function Documentation

5.1.3.1 void LCD::bin (int *n*)

shows an integer in binary format

5.1.3.2 void LCD::blink (bool *set*)

Set blink mode.

5.1.3.3 void LCD::clean ( )

clean the LCD screen

5.1.3.4 void LCD::dec (int *n*)

shows an integer in decimal format

5.1.3.5 void LCD::enable (bool *s*)

Set the display on or off.

5.1.3.6 void LCD::error (String *m*)

shows an error message

5.1.3.7 void LCD::error (String *m*, int *x*, int *y*)

shows an error message at specified coordinates

5.1.3.8 void LCD::hex (int *n*)

shows an integer in hexadeciaml format

5.1.3.9 void LCD::menu (String *sect1*, String *sect2*, String *sect3*, String *sect4*)

5.1.3.10 void LCD::message (String *m*)

shows a string message

5.1.3.11 void LCD::message (String *m*, int *x*, int *y*)

shows a string message at specified coordinates

5.1.3.12 void LCD::oct (int *n*)

shows an integer in octal format

5.1.3.13 LCD& LCD::operator= (const LCD & *c*) [private]

5.1.3.14 void LCD::welcome ( )

shows the program welcome message

5.1.4 Member Data Documentation

5.1.4.1 AlphaLCD LCD::lcd [private]

AlphaLCD class inherited instance.

Definition at line 53 of file LCD.h.

The documentation for this class was generated from the following file:

- /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit_serial_pi/Meditech_-ChipKitControlPanel/LCD.h

5.2 Temperature Class Reference

```
#include "Temperature.h"
```

Public Member Functions

- Temperature ()

  Temperature class constructor.
- void CalcTemp (float sensor)

  Convert the analog value to the respective value.
- float Celsius () const

Private Attributes

- float _Celsius
- float _sensorValue

5.2.1 Detailed Description

Definition at line 17 of file Temperature.h.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Temperature::Temperature ( )

Temperature class constructor.

Set the analog pin for data reading

Definition at line 14 of file Temperature.cpp.

```
14                              {
15  }
```

5.2.3 Member Function Documentation

5.2.3.1 void Temperature::CalcTemp (float *sensor*)

Convert the analog value to the respective value.

As the call to the temperature reading is not a high priority task and does not need very high responsivity, when the analog data are read they are immediately converted to the three temperature scales: Celsius, Kelvin and Farehneit

Definition at line 24 of file Temperature.cpp.

References _Celsius, and _sensorValue.

Referenced by loop(), setup(), and tempMonitor().

```
24                                          {
25
26    _sensorValue = sensor;
27    _Celsius = (((_sensorValue / 1024) * 5) * 100) - 10; // convert the reading to C
28  }
```

5.2.3.2 float Temperature::Celsius ( ) const

Definition at line 30 of file Temperature.cpp.

References _Celsius.

Referenced by loop(), temperature(), and tempMonitor().

```
30                                  {
31     return(_Celsius);
32 }
```

5.2.4 Member Data Documentation

5.2.4.1 float Temperature::_Celsius [private]

Definition at line 24 of file Temperature.h.

Referenced by CalcTemp(), and Celsius().

5.2.4.2 float Temperature::_sensorValue [private]

Definition at line 25 of file Temperature.h.

Referenced by CalcTemp().

The documentation for this class was generated from the following files:

- /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit_serial_pi/Meditech_-ChipKitControlPanel/Temperature.h
- /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit_serial_pi/Meditech_-ChipKitControlPanel/Temperature.cpp

Chapter 6

File Documentation

6.1 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/Analog.h File Reference

Analog input constants.

Macros

- #define KOHM 0

  Units to show the potentiometer resistance value. Set to 0 if the resistance should be in Ohm instead of K.
- #define ANALOGPOT 4700.00

  Analog potentiometer value in Ohm.
- #define ANALOGDIVIDER 1024.00

  Steps divider from analog converter.
- #define MINGAIN 16

  Minimum gain level mapped to the analog reading.
- #define MAXGAIN 64
- #define CALIBRATION_POT A0

6.1.1 Detailed Description

Analog input constants. Constants definitions to manage the analog potentiometer input for digital stethoscope calibration

Definition in file Analog.h.

6.1.2 Macro Definition Documentation

6.1.2.1 #define ANALOGDIVIDER 1024.00

Steps divider from analog converter.

Definition at line 17 of file Analog.h.

Referenced by loop().

6.1.2.2 #define ANALOGPOT 4700.00

Analog potentiometer value in Ohm.

Definition at line 15 of file Analog.h.

6.1.2.3 #define CALIBRATION_POT A0

Definition at line 23 of file Analog.h.

Referenced by loop().

6.1.2.4 #define KOHM 0

Units to show the potentiometer resistance value. Set to 0 if the resistance should be in Ohm instead of K.

Definition at line 13 of file Analog.h.

6.1.2.5 #define MAXGAIN 64

Definition at line 21 of file Analog.h.

Referenced by loop().

6.1.2.6 #define MINGAIN 16

Minimum gain level mapped to the analog reading.

Definition at line 20 of file Analog.h.

Referenced by loop().

6.2 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/Globals.h File Reference

Global constants.

Macros

- #define LIDOPEN false
  Lid open status.
- #define LIDCLOSED true
  Lid closed status.
- #define FAN_SPEED 2
  PWM fan control speed PIN.
- #define ECG_STATUS 3
  ECG enabled PIN.
- #define STETHOSCOPE_STATUS 8
  Stethoscope enabled PIN.
- #define CAMERA_STATUS 9
  Camera transmitting enabled PIN.
- #define LIDSTATUS 10
  Lid switch input pin.

- #define LCDdataPin 11

  LCD Shift control pin - Data signal Define this value accordingly with the available I2C board pins.
- #define LCDclockPin 12

  LCD Shift control pin - Clock signal Define this value accordingly with the available I2C board pins.
- #define LCDlatchPin 13

  LCD Shift control pin - Latch signal Define this value accordingly with the available I2C board pins.

6.2.1 Detailed Description

Global constants.

Definition in file Globals.h.

6.2.2 Macro Definition Documentation

6.2.2.1 #define CAMERA_STATUS 9

Camera transmitting enabled PIN.

Definition at line 21 of file Globals.h.

Referenced by setup(), and testStatusLED().

6.2.2.2 #define ECG_STATUS 3

ECG enabled PIN.

Definition at line 17 of file Globals.h.

Referenced by setup(), and testStatusLED().

6.2.2.3 #define FAN_SPEED 2

PWM fan control speed PIN.

Definition at line 15 of file Globals.h.

6.2.2.4 #define LCDclockPin 12

LCD Shift control pin - Clock signal Define this value accordingly with the available I2C board pins.

Definition at line 30 of file Globals.h.

6.2.2.5 #define LCDdataPin 11

LCD Shift control pin - Data signal Define this value accordingly with the available I2C board pins.

Definition at line 27 of file Globals.h.

6.2.2.6 #define LCDlatchPin 13

LCD Shift control pin - Latch signal Define this value accordingly with the available I2C board pins.

Definition at line 33 of file Globals.h.

6.2.2.7 #define LIDCLOSED true

Lid closed status.

Definition at line 12 of file Globals.h.

Referenced by loop(), and setup().

6.2.2.8 #define LIDOPEN false

Lid open status.

Definition at line 10 of file Globals.h.

Referenced by loop().

6.2.2.9 #define LIDSTATUS 10

Lid switch input pin.

Definition at line 23 of file Globals.h.

Referenced by isLidClose(), and setup().

6.2.2.10 #define STETHOSCOPE_STATUS 8

Stethoscope enabled PIN.

Definition at line 19 of file Globals.h.

Referenced by setup(), and testStatusLED().

6.3 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/LCD.h File Reference

LCD display Manager include file.

```
#include <inttypes.h>
#include <Print.h>
#include <AlphaLCD.h>
#include <Streaming.h>
```

Classes

- class LCD

  Manages the Alphanumeric display for program output messages.

Macros

- #define LCDCHARS 20

  Display characters per line Define this value accordingly with the LCD Hardware datasheet.
- #define LCDROWS 2

  Display rows.
- #define LCDTOPROW 0

  The top row number of the LCD.

- #define LCDBOTTOMROW 1

  The bottom row number of the LCD.
- #define LCD_SECTOR1 0

  Top Left display sector column.
- #define LCD_SECTOR2 LCDCHARS / 2

  Top Right display sector column (usually are always pair numbers)
- #define LCD_SECTOR3 0

  Bottom Left display sector column.
- #define LCD_SECTOR4 LCDCHARS / 2

  Bottom Right display sector column.
- #define LCDERROR_DELAY 5000

  Delay after showing an error.
- #define LCDMESSAGE_DELAY 5000

  Delay after showing a temporary message e.g. the welcome screen.
- #define LCDCLEAR_DELAY 50

  Delay after a clear display call to hardware has been done.

6.3.1 Detailed Description

LCD display Manager include file. Methods to manage the LCD output and display features, including some hard-coded strings that should be present, like the welcome message.

Definition in file LCD.h.

6.3.2 Macro Definition Documentation

6.3.2.1 #define LCD_SECTOR1 0

Top Left display sector column.

Definition at line 27 of file LCD.h.

Referenced by menu().

6.3.2.2 #define LCD_SECTOR2 LCDCHARS / 2

Top Right display sector column (usually are always pair numbers)

Definition at line 29 of file LCD.h.

Referenced by menu().

6.3.2.3 #define LCD_SECTOR3 0

Bottom Left display sector column.

Definition at line 31 of file LCD.h.

Referenced by menu().

6.3.2.4 #define LCD_SECTOR4 LCDCHARS / 2

Bottom Right display sector column.

Definition at line 33 of file LCD.h.

Referenced by menu().

6.3.2.5 #define LCDBOTTOMROW 1

The bottom row number of the LCD.

Definition at line 25 of file LCD.h.

Referenced by menu(), stethoscopeGainLevel(), stethoscopeMsg(), and welcome().

6.3.2.6 #define LCDCHARS 20

Display characters per line Define this value accordingly with the LCD Hardware datasheet.

Definition at line 19 of file LCD.h.

Referenced by setup().

6.3.2.7 #define LCDCLEAR_DELAY 50

Delay after a clear display call to hardware has been done.

Definition at line 40 of file LCD.h.

Referenced by clean().

6.3.2.8 #define LCDERROR_DELAY 5000

Delay after showing an error.

Definition at line 36 of file LCD.h.

Referenced by error().

6.3.2.9 #define LCDMESSAGE_DELAY 5000

Delay after showing a temporary message e.g. the welcome screen.

Definition at line 38 of file LCD.h.

Referenced by welcome().

6.3.2.10 #define LCDROWS 2

Display rows.

Definition at line 21 of file LCD.h.

Referenced by setup().

6.3.2.11 #define LCDTOPROW 0

The top row number of the LCD.

Definition at line 23 of file LCD.h.

Referenced by loop(), menu(), stethoscopeMsg(), and welcome().

6.4 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit_serial_pi/Meditech_ChipKitControlPanel/Meditech_ChipKitControlPanel.pde File Reference

meditech ChipKit control panel main application

```
#include <AlphaLCD.h>
#include <Streaming.h>
#include "LCD.h"
#include "Analog.h"
#include "Strings.h"
#include "Version.h"
#include "Globals.h"
#include "Temperature.h"
#include <SoftPWMServo.h>
```

Functions

- AlphaLCD lcd (11, 12, 13)
  Display class instance.
- void setup ()
  Initialisation method.
- void loop (void)
  Main loop application.
- void tempMonitor ()
  Continuously monitor the internal temperature.
- void temperature ()
  shows the actual internal temperature
- boolean isLidClose ()
  Check for lid status.
- void stethoscopeMsg ()
  Stethoscope title.
- void stethoscopeGainLevel (double value)
  Update the scaled analog value corresponding to the gain setting.
- void welcome ()
  Welcome message shown at device power-on.
- void message (String m)
  Display a string on the LCD at the cursor position.
- void dec (int n)
  Display an integer value in decimal format at the cursor position.
- void hex (int n)
  Display an integer value in hex format at the cursor position.
- void error (String m, int x, int y)
  Display an error message at the specified cursor coordinates.
- void error (String m)
  Display an error message at the cursor position.
- void message (String m, int x, int y)
  Display a string on the LCD at the specified cursor coordinates.
- void clean ()
  Clean the display.
- void menu (String sect1, String sect2, String sect3, String sect4)
  Creates a menu screen.
- void testStatusLED ()

Variables

- Temperature internalTemp

  Internal temperature sensor class instance.
- int pValue = 0

  Initial potentiometer value.
- boolean lidStatus

  Current lid status.

6.4.1 Detailed Description

meditech ChipKit control panel main application

Definition in file Meditech_ChipKitControlPanel.pde.

6.4.2 Function Documentation

6.4.2.1 void clean ( )

Clean the display.

A delay of 100 ms is added after the hardware clear() call to give the display the time to complete the operation.

Definition at line 297 of file Meditech_ChipKitControlPanel.pde.

References lcd(), and LCDCLEAR_DELAY.

Referenced by menu().

```
297             {
298   lcd.clear();
299   delay(LCDCLEAR_DELAY);
300 }
```

6.4.2.2 void dec (int *n*)

Display an integer value in decimal format at the cursor position.

Parameters

n	the integer to show in decimal format

Definition at line 233 of file Meditech_ChipKitControlPanel.pde.

References lcd().

```
233               {
234   lcd.print(n, DEC);
235 }
```

6.4.2.3 void error (String *m*, int *x*, int *y*)

Display an error message at the specified cursor coordinates.

The error message is shown for a LCDERROR_DELAY milliseconds. After the timeout expires the screen is not cleared so the next steps should be managed by the program flow. It is expected that error messages are shown in a calling code that manages the error conditions.

Parameters

m	the message string
x	the cursor column zero based
y	the row number zero based

Definition at line 259 of file Meditech_ChipKitControlPanel.pde.

References LCDERROR_DELAY, and message().

```
259                                     {
260   message(m, x, y);
261   delay(LCDERROR_DELAY);
262 }
```

6.4.2.4 void error (String *m*)

Display an error message at the cursor position.

The error message is shown for a LCDERROR_DELAY milliseconds. After the timeout expires the screen is not cleared so the next steps should be managed by the program flow. It is expected that error messages are shown in a calling code that manages the error conditions.

Parameters

m	the string message

Definition at line 274 of file Meditech_ChipKitControlPanel.pde.

References LCDERROR_DELAY, and message().

```
274                       {
275   message(m);
276   delay(LCDERROR_DELAY);
277 }
```

6.4.2.5 void hex (int *n*)

Display an integer value in hex format at the cursor position.

Parameters

n	the integer to show in hexadecimal format

Definition at line 242 of file Meditech_ChipKitControlPanel.pde.

References lcd().

```
242                   {
243   lcd.print("0x");
244   lcd.print(n, HEX);
245 }
```

6.4.2.6 boolean isLidClose ( )

Check for lid status.

Definition at line 173 of file Meditech_ChipKitControlPanel.pde.

References LIDSTATUS.

Referenced by loop().

```
173                       {
174    return digitalRead(LIDSTATUS);
175  }
```

6.4.2.7 AlphaLCD lcd (11 , 12 , 13)

Display class instance.

Referenced by clean(), dec(), hex(), loop(), message(), setup(), stethoscopeGainLevel(), stethoscopeMsg(), temperature(), tempMonitor(), and welcome().

6.4.2.8 void loop (void)

Main loop application.

Definition at line 93 of file Meditech_ChipKitControlPanel.pde.

References _LID_OPEN, ANALOGDIVIDER, Temperature::CalcTemp(), CALIBRATION_POT, Temperature::Celsius(), isLidClose(), lcd(), LCDTOPROW, LIDCLOSED, LIDOPEN, lidStatus, MAXGAIN, message(), MINGAIN, pValue, stethoscopeGainLevel(), stethoscopeMsg(), TEMP_SENSOR, tempMonitor(), and testStatusLED().

```
94 {
95   #ifdef _DEBUG
96   Serial1 << "=== DEBUG ON ==" << endl;
97   testFan();
98   // Read the actual temperature
99   internalTemp.CalcTemp(analogRead(TEMP_SENSOR));
100   Serial1 << "Temperature read: " << internalTemp.Celsius() << endl;
101   Serial1 << "=== DEBUG OFF ==" << endl;
102   #endif
103   // Check if the lid is open
104   if(isLidClose() == LIDCLOSED) {
105     // If the lid was open before, reset the display status
106     if(lidStatus == LIDOPEN) {
107       // Reset the status
108       lidStatus = LIDCLOSED;
109 
110       // Digital stethoscope calibration
111       stethoscopeMsg();
112     }
113 
114     pValue = analogRead(CALIBRATION_POT);  // Read the pot value
115 
116     int gainValue = map(pValue, 0, ANALOGDIVIDER, MINGAIN,
    MAXGAIN);
117 
118     // Update the display
119     stethoscopeGainLevel(gainValue);
120 
121   }
122   else {
123       // Set the status flag
124       lidStatus = LIDOPEN;
125 
126       // Show the error message
127       lcd.clear();
128       message(_LID_OPEN, 5, LCDTOPROW);
129 
130       // Temperature monitoring
131       tempMonitor();
132 
133       delay(50);
134 
135       testStatusLED();
136 
137   }
138 }
```

6.4.2.9 void menu (String *sect1*, String *sect2*, String *sect3*, String *sect4*)

Creates a menu screen.

The 2-lines LCD screen is divided in four sectors, that can be used or not. The LCD sectors length and position are defined and based on the LCD size. Every sector is filled with one of the four parameters string. Sectors 1 & 2 are in the top row, sectors 3 & 4 in the bottom row

Parameters

sect1	The Upper Left display sector
sect2	The Upper Right display sector
sect3	The Lower Left display sector
sect4	The Lower Right display sector

Definition at line 315 of file Meditech_ChipKitControlPanel.pde.

References clean(), LCD_SECTOR1, LCD_SECTOR2, LCD_SECTOR3, LCD_SECTOR4, LCDBOTTOMROW, LCDTOPROW, and message().

```
315                                                                 {
316   clean();
317   message(sect1, LCD_SECTOR1, LCDTOPROW);
318   message(sect2, LCD_SECTOR2, LCDTOPROW);
319   message(sect3, LCD_SECTOR3, LCDBOTTOMROW);
320   message(sect4, LCD_SECTOR4, LCDBOTTOMROW);
321 }
```

6.4.2.10 void message (String *m*)

Display a string on the LCD at the cursor position.

Parameters

m	the message string

Definition at line 224 of file Meditech_ChipKitControlPanel.pde.

References lcd().

Referenced by error(), loop(), menu(), and message().

```
224                         {
225   lcd.print(m);
226 }
```

6.4.2.11 void message (String *m*, int *x*, int *y*)

Display a string on the LCD at the specified cursor coordinates.

Parameters

m	the string message
x	the cursor column zero based
y	the row number zero based

Definition at line 286 of file Meditech_ChipKitControlPanel.pde.

References lcd(), and message().

```
286                                     {
287   lcd.setCursor(x, y);
288   message(m);
289 }
```

6.4.2.12 void setup ( )

Initialisation method.

Definition at line 54 of file Meditech_ChipKitControlPanel.pde.

References Temperature::CalcTemp(), CAMERA_STATUS, ECG_STATUS, lcd(), LCDCHARS, LCDROWS, LIDCLOSED, LIDSTATUS, lidStatus, STETHOSCOPE_STATUS, stethoscopeMsg(), TEMP_SENSOR, temperature(), and welcome().

```
55 {
56
57   #ifdef _DEBUG
58   Serial1.begin(9600);
59   #endif
60
61   // Initializes the Lid status swith pin
62   pinMode(LIDSTATUS, INPUT);
63   lidStatus = LIDCLOSED;
64
65   pinMode(ECG_STATUS, OUTPUT);
66   pinMode(STETHOSCOPE_STATUS, OUTPUT);
67   pinMode(CAMERA_STATUS, OUTPUT);
68
69   digitalWrite(ECG_STATUS, HIGH);
70   digitalWrite(STETHOSCOPE_STATUS, HIGH);
71   digitalWrite(CAMERA_STATUS, HIGH);
72
73   // Initializes the LCD library
74   lcd.begin(LCDCHARS, LCDROWS);
75
76   // Turn LCD On
77   lcd.display();
78
79   // Initialisation string
80   welcome();
81
82   // Read the actual temperature
83   internalTemp.CalcTemp(analogRead(TEMP_SENSOR));
84
85   // Shows the actual internal temperature
86   temperature();
87
88   // Digital stethoscope calibration
89   stethoscopeMsg();
90 }
```

6.4.2.13 void stethoscopeGainLevel (double *value*)

Update the scaled analog value corresponding to the gain setting.

Definition at line 193 of file Meditech_ChipKitControlPanel.pde.

References lcd(), and LCDBOTTOMROW.

Referenced by loop().

```
193                                         {
194   lcd.setCursor(12, LCDBOTTOMROW);
195   lcd << value;
196 }
```

6.4.2.14 void stethoscopeMsg ( )

Stethoscope title.

Definition at line 180 of file Meditech_ChipKitControlPanel.pde.

References lcd(), LCDBOTTOMROW, and LCDTOPROW.

Referenced by loop(), and setup().

```
180                       {
181
182   lcd.clear();
183   lcd.setCursor(0, LCDTOPROW);
184   lcd << _STET;
185   lcd.setCursor(0, LCDBOTTOMROW);
186   lcd << _STET_CAL;
187
188 }
```

6.4.2.15 void temperature ( )

shows the actual internal temperature

Definition at line 159 of file Meditech_ChipKitControlPanel.pde.

References _CELSIUS, _EMPTY_NUMBER4, _INTERNAL_TEMP, Temperature::Celsius(), and lcd().

Referenced by setup().

```
159                   {
160
161   lcd.setCursor(0, 0);
162   lcd.print(_INTERNAL_TEMP);
163   lcd.setCursor(0, 1);
164   lcd  << _EMPTY_NUMBER4 << internalTemp.Celsius() <<
      _CELSIUS;
165   delay(2500);
166   lcd.clear();
167
168 }
```

6.4.2.16 void tempMonitor ( )

Continuously monitor the internal temperature.

Definition at line 145 of file Meditech_ChipKitControlPanel.pde.

References _CELSIUS, _INTERNAL_TEMP, Temperature::CalcTemp(), Temperature::Celsius(), lcd(), and TEMP_-SENSOR.

Referenced by loop().

```
145                   {
146
147   // Read the actual temperature
148   internalTemp.CalcTemp(analogRead(TEMP_SENSOR));
149
150   lcd.setCursor(0, 1);
151   lcd << _INTERNAL_TEMP << " " << internalTemp.
      Celsius() << _CELSIUS;
152   delay(50);
153
154 }
```

6.4.2.17 void testStatusLED ( )

Definition at line 324 of file Meditech_ChipKitControlPanel.pde.

References CAMERA_STATUS, ECG_STATUS, and STETHOSCOPE_STATUS.

Referenced by loop().

```
324                     {
325
326   digitalWrite(ECG_STATUS, LOW);
327   delay(200);
328   digitalWrite(STETHOSCOPE_STATUS, LOW);
329   delay(200);
330   digitalWrite(CAMERA_STATUS, LOW);
331   delay(200);
332
333   digitalWrite(ECG_STATUS, HIGH);
334   delay(200);
335   digitalWrite(STETHOSCOPE_STATUS, HIGH);
336   delay(200);
337   digitalWrite(CAMERA_STATUS, HIGH);
338   delay(200);
339
340 }
```

6.4.2.18 void welcome ( )

Welcome message shown at device power-on.

Definition at line 201 of file Meditech_ChipKitControlPanel.pde.

References _BD, _BUILD, _MEDITECH, _SPACING, _VERSION, build, lcd(), LCDBOTTOMROW, LCDMESSAGE_DELAY, LCDTOPROW, project, and version.

Referenced by setup().

```
201               {
202
203   lcd.clear();
204   lcd.setCursor(0, LCDTOPROW);
205   lcd << project();
206   lcd.setCursor(0, LCDBOTTOMROW);
207   lcd << _VERSION << _SPACING << version() << _SPACING <<
      _BUILD << _SPACING << build();
208   delay(LCDMESSAGE_DELAY);
209   lcd.clear();
210
211   lcd.setCursor(0, 0);
212   lcd.print(_BD);
213   lcd.setCursor(0, 1);
214   lcd.print(_MEDITECH);
215   delay(2500);
216   lcd.clear();
217 }
```

6.4.3 Variable Documentation

6.4.3.1 Temperature internalTemp

Internal temperature sensor class instance.

Definition at line 45 of file Meditech_ChipKitControlPanel.pde.

6.4.3.2 boolean lidStatus

Current lid status.

Definition at line 51 of file Meditech_ChipKitControlPanel.pde.

Referenced by loop(), and setup().

6.4.3.3 int pValue = 0

Initial potentiometer value.

Definition at line 48 of file Meditech_ChipKitControlPanel.pde.

Referenced by loop().

6.5 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/Strings.h File Reference

LCD Display base strings.

```
#include "Version.h"
```

Macros

- #define _SPACING " "
- #define _EMPTY_HALF_LINE " "
- #define _EMPTY_NUMBER4 " "
- #define _BD "Balearic Dynamics"
- #define _MEDITECH "Meditech beta"
- #define _VERSION "Ver"
- #define _BUILD "Bld"
- #define _INTERNAL_TEMP "Int.temp. "
- #define _CELSIUS "C"
- #define _LID_OPEN "Lid Open !!!"

6.5.1 Detailed Description

LCD Display base strings. The locally defined strings refers only to the automated high-priority messages while the probe messages and the application information are sent from the PI master controlling device.

Definition in file Strings.h.

6.5.2 Macro Definition Documentation

6.5.2.1 #define _BD "Balearic Dynamics"

Definition at line 20 of file Strings.h.

Referenced by welcome().

6.5.2.2 #define _BUILD "Bld"

Definition at line 23 of file Strings.h.

Referenced by welcome().

6.5.2.3 #define _CELSIUS "C"

Definition at line 26 of file Strings.h.

Referenced by temperature(), and tempMonitor().

6.5.2.4 #define _EMPTY_HALF_LINE " "

Definition at line 17 of file Strings.h.

6.5.2.5 #define _EMPTY_NUMBER4 " "

Definition at line 18 of file Strings.h.

Referenced by temperature().

6.5.2.6 #define _INTERNAL_TEMP "Int.temp. "

Definition at line 25 of file Strings.h.

Referenced by temperature(), and tempMonitor().

6.5.2.7 #define _LID_OPEN "Lid Open !!!"

Definition at line 28 of file Strings.h.

Referenced by loop().

6.5.2.8 #define _MEDITECH "Meditech beta"

Definition at line 21 of file Strings.h.

Referenced by welcome().

6.5.2.9 #define _SPACING " "

Definition at line 16 of file Strings.h.

Referenced by welcome().

6.5.2.10 #define _VERSION "Ver"

Definition at line 22 of file Strings.h.

Referenced by welcome().

6.6 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/Temperature.cpp File Reference

Class to manage the internal temperature of the devices.

```
#include "Temperature.h"
```

6.6.1 Detailed Description

Class to manage the internal temperature of the devices. The Temperature class is based on the LM35 temperature sensor

Definition in file Temperature.cpp.

6.7 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit-_serial_pi/Meditech_ChipKitControlPanel/Temperature.h File Reference

Constants and class prototypes.

```
#include <inttypes.h>
```

Classes

- class Temperature

Macros

- #define KELVINC 273

  Constant for Kelvin temperature conversion.
- #define TEMP_SENSOR A1

  Analog channel for the temperature sensor.

6.7.1 Detailed Description

Constants and class prototypes.

Definition in file Temperature.h.

6.7.2 Macro Definition Documentation

6.7.2.1 #define KELVINC 273

Constant for Kelvin temperature conversion.

Definition at line 12 of file Temperature.h.

6.7.2.2 #define TEMP_SENSOR A1

Analog channel for the temperature sensor.

Definition at line 15 of file Temperature.h.

Referenced by loop(), setup(), and tempMonitor().

6.8 /Volumes/John Doe/Development Projects/Meditech SciFi/SourcesRepositories/chipkit_serial_pi/Meditech_ChipKitControlPanel/Version.h File Reference

Version and Build Number Helper Class.

Macros

- #define build() "0009"

  Incremental build number.
- #define version() "1.0"

  Firmware version.
- #define project() "Meditech beta"

  Project name.

6.8.1 Detailed Description

Version and Build Number Helper Class. This helper macros exposes the static methods to get the firmware version and the build number. Use the build() and version() methods anywhere in the program including this file

Definition in file Version.h.

6.8.2 Macro Definition Documentation

6.8.2.1 #define build( ) "0009"

Incremental build number.

Definition at line 12 of file Version.h.

Referenced by welcome().

6.8.2.2 #define project( ) "Meditech beta"

Project name.

Definition at line 16 of file Version.h.

Referenced by welcome().

6.8.2.3 #define version( ) "1.0"

Firmware version.

Definition at line 14 of file Version.h.

Referenced by welcome().

Index