Travail Pratique Gestion de Train

1

Generated by Doxygen 1.7.6.1

Thu Jun 19 2014 19:45:54

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 button_t Struct Reference

`#include <button.h>`

Data Fields

- char name [20]
- uint16_t x_start
- uint16_t x_stop
- uint16_t y_start
- uint16_t y_stop

3.1.1 Field Documentation

3.1.1.1 char name[20]

3.1.1.2 uint16_t x_start

3.1.1.3 uint16_t x_stop

3.1.1.4 uint16_t y_start

3.1.1.5 uint16_t y_stop

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

- button.h

Chapter 4

File Documentation

4.1 button.c File Reference

Permit to add buttons (virtuals) and to test when we.

`#include "button.h"` Include dependency graph for button.c:

Functions

- char * IsClickOnAButton (uint16_t x, uint16_t y)
- void AddButton (char *button_name, uint16_t x_start, uint16_t x_stop, uint16_t y_start, uint16_t y_stop)

Variables

- button_t buttons [NB_BUTTONS]
- int nb_buttons_available = -1

4.1.1 Detailed Description

Permit to add buttons (virtuals) and to test when we.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 press on the screen if the preassure is on a button.

4.1.2 Function Documentation

4.1.2.1 void AddButton (char * *button_name*, uint16_t *x_start*, uint16_t *x_stop*, uint16_t *y_start*, uint16_t *y_stop*)

Add a new button to the list

Parameters

button_name	String that contain button name
x_start	Where the button start on X
x_stop	Where the button start on X
y_start	Where the button start on Y
y_stop	Where the button stop on Y

4.1.2.2 char* IsClickOnAButton (uint16_t *x*, uint16_t *y*)

Permit to test if a button is present where the pressure occurred

Parameters

x	Value of x where the pressure occurred
y	Value of y where the pressure occurred

4.1.3 Variable Documentation

4.1.3.1 button_t buttons[NB_BUTTONS]

4.1.3.2 int nb_buttons_available = -1

4.2 button.h File Reference

Permit to add buttons (virtuals) and to test when we.

`#include "string.h"` `#include "LPC17xx.h"` Include dependency graph for button.h: This graph shows which files directly or indirectly include this file:

Data Structures

- struct button_t

Defines

- #define NB_BUTTONS 10

Functions

- void AddButton (char *button_name, uint16_t x_start, uint16_t x_stop, uint16_t y_start, uint16_t y_stop)
- char * IsClickOnAButton (uint16_t x, uint16_t y)

4.2.1 Detailed Description

Permit to add buttons (virtuals) and to test when we.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 press on the screen if the preassure is on a button.

4.2.2 Define Documentation

4.2.2.1 #define NB_BUTTONS 10

4.2.3 Function Documentation

4.2.3.1 void AddButton (char * *button_name,* uint16_t *x_start,* uint16_t *x_stop,* uint16_t *y_start,* uint16_t *y_stop*)

Add a new button to the list

Parameters

button_name	String that contain button name
x_start	Where the button start on X
x_stop	Where the button start on X
y_start	Where the button start on Y
y_stop	Where the button stop on Y

4.2.3.2 char* IsClickOnAButton (uint16_t *x,* uint16_t *y*)

Permit to test if a button is present where the pressure occurred

Parameters

x	Value of x where the pressure occurred
y	Value of y where the pressure occurred

4.3 drawScreen.c File Reference

Contain functions to draws things on the screen.

`#include "drawScreen.h"` Include dependency graph for drawScreen.c:

Functions

- void change_zone_color (uint16_t x_start, uint16_t x_stop, uint16_t y_start, uint16_t y_stop, uint8_t ∗color_character)
- void draw_arrow_right (uint16_t x_start, uint16_t y_start, uint8_t thickness, uint8_t height_arrow, uint8_t length, uint8_t ∗color)
- void draw_arrow_left (uint16_t x_start, uint16_t y_start, uint8_t thickness, uint8_t height_arrow, uint8_t length, uint8_t ∗color)
- void display_lights (uint8_t ∗color, uint8_t length)

4.3.1 Detailed Description

Contain functions to draws things on the screen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.3.2 Function Documentation

4.3.2.1 void change_zone_color (uint16_t *x_start*, uint16_t *x_stop*, uint16_t *y_start*, uint16_t *y_stop*, uint8_t ∗ *color_character*)

Change a color in a zone

Parameters

x_start	Where start x in the screen
x_stop	Where stop x in the screen
y_start	Where start y in the screen
y_stop	Where stop y in the screen
color_characted	Which color the zone is changed [Red, Green, Blue]

4.3.2.2 void display_lights (uint8_t * *color*, uint8_t *length*)

Draw 6 sunbeams around the lamp to show the lights on

Parameters

color	Sunbeams color [Red, Green, Blue]
length	Length of the sunbeams

4.3.2.3 void draw_arrow_left (uint16_t *x_start*, uint16_t *y_start*, uint8_t *thickness*, uint8_t *height_arrow*, uint8_t *length*, uint8_t * *color*)

Draw an arrow to the left

Parameters

x_start	Where the arrow tip start on x
y_start	Where the arrow tip start on y
thickness	Thickness of the arrow
height_arrow	Arrow's height
color	Arrow color [Red, Green, Blue]

4.3.2.4 void draw_arrow_right (uint16_t *x_start*, uint16_t *y_start*, uint8_t *thickness*, uint8_t *height_arrow*, uint8_t *length*, uint8_t * *color*)

Draw an arrow to the right

Parameters

x_start	Where the arrow base start on x
y_start	Where the arrow base start on y
thickness	Thickness of the arrow
height_arrow	Arrow's height
color	Arrow color [Red, Green, Blue]

4.4 drawScreen.h File Reference

`#include "LPC17xx.h"` `#include "ExtLab2.h"` `#include "Ecran.h"` `#include <cr_section_macros.h>` Include dependency graph for drawScreen.h: This graph shows which files directly or indirectly include this file:

Functions

- void change_zone_color (uint16_t x_start, uint16_t x_stop, uint16_t y_start, uint16_t y_stop, uint8_t ∗color_character)
- void draw_arrow_right (uint16_t x_start, uint16_t y_start, uint8_t thickness, uint8_t heigth_arrow, uint8_t length, uint8_t ∗color)
- void draw_arrow_left (uint16_t x_start, uint16_t y_start, uint8_t thickness, uint8_t heigth_arrow, uint8_t length, uint8_t ∗color)
- void display_lights (uint8_t ∗color, uint8_t length)

4.4.1 Detailed Description

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 Contain functions to draws things on the screen.

4.4.2 Function Documentation

4.4.2.1 void change_zone_color (uint16_t *x_start,* uint16_t *x_stop,* uint16_t *y_start,* uint16_t *y_stop,* uint8_t ∗ *color_character*)

Change a color in a zone

Parameters

x_start	Where start x in the screen
x_stop	Where stop x in the screen
y_start	Where start y in the screen
y_stop	Where stop y in the screen
color_characted	Which color the zone is changed [Red, Green, Blue]

4.4.2.2 void display_lights (uint8_t * *color,* uint8_t *length*)

Draw 6 sunbeams around the lamp to show the lights on

Parameters

color	Sunbeams color [Red, Green, Blue]
length	Length of the sunbeams

4.4.2.3 void draw_arrow_left (uint16_t *x_start,* uint16_t *y_start,* uint8_t *thickness,* uint8_t *height_arrow,* uint8_t *length,* uint8_t * *color*)

Draw an arrow to the left

Parameters

x_start	Where the arrow tip start on x
y_start	Where the arrow tip start on y
thickness	Thickness of the arrow
height_arrow	Arrow's height
color	Arrow color [Red, Green, Blue]

4.4.2.4 void draw_arrow_right (uint16_t *x_start,* uint16_t *y_start,* uint8_t *thickness,* uint8_t *height_arrow,* uint8_t *length,* uint8_t * *color*)

Draw an arrow to the right

Parameters

x_start	Where the arrow base start on x
y_start	Where the arrow base start on y
thickness	Thickness of the arrow
height_arrow	Arrow's height
color	Arrow color [Red, Green, Blue]

4.5 Ecran.c File Reference

Contain functions to control the screen.

`#include "Ecran.h"` Include dependency graph for Ecran.c:

Functions

- void Init_ports_display ()
- void Index_out (uint8_t idx)
- void Parameter_out (uint16_t param)
- void Set_gamma ()
- void Send_color (uint8_t color)
- void Init_display ()
- void Write_pixel (uint8_t red, uint8_t green, uint8_t blue)
- void Set_cursor (uint16_t x, uint16_t y)
- void Create_partial_screen (uint16_t v_start, uint16_t v_end, uint8_t h_start, uint8_t h_end)

4.5.1 Detailed Description

Contain functions to control the screen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.5.2 Function Documentation

4.5.2.1 void Create_partial_screen (uint16_t *v_start,* uint16_t *v_end,* uint8_t *h_start,* uint8_t *h_end*)

Creation of a window on the screen

create a window on the screen by precising a size

Parameters

v_start	largeur = v_start to v_end
v_end	
h_start	longueur = h_start to h_end
h_end	

4.5.2.2 void Index_out (uint8_t *idx*)

Permit to choose on which index we want to write on the screen options.

Parameters

idx	: Index address to access

4.5.2.3 void Init_display ( )

Screen initialisation

Set configuration: 8bits,262k color,CPU interface,Standby OFF,Set Gamma,Display on

4.5.2.4 void Init_ports_display ( )

Initialize all GPIO used with the screen.

4.5.2.5 void Parameter_out (uint16_t *param*)

Permit to change parameters on the index selected with the function Index_out.

Parameters

param	: Datas to write

4.5.2.6 void Send_color (uint8_t *color*)

Send a color to the touchscreen, this function need to be called 3 times to set all color on a pixel (R, G, B)

Parameters

color	Color sent

4.5.2.7 void Set_cursor (uint16_t *x*, uint16_t *y*)

Set cursor localisation

Parameters

x	value between 0 and 240
y	value between 0 and 320

4.5.2.8 void Set_gamma ( )

Set gamma of the screen

Set register value, to fix color on th screen

4.5.2.9 void Write_pixel (uint8_t *red*, uint8_t *green*, uint8_t *blue*)

send RGB color on pixel

Parameters

red	value between 0 and 255
green	value between 0 and 255
blue	value between 0 and 255

4.6 Ecran.h File Reference

`#include "LPC17xx.h"` `#include "ExtLab2.h"` `#include <cr_section_macros.h>` Include dependency graph for Ecran.h: This graph shows which files directly or indirectly include this file:

Defines

- #define DISPLAY_CS 19
- #define DISPLAY_RS 18
- #define DISPLAY_WRB 20
- #define LCD_WIDTH 240
- #define LCD_HEIGHT 320
- #define SIZE_LCD 76800

Functions

- void Init_ports_display ()
- void Index_out (uint8_t idx)
- void Parameter_out (uint16_t param)
- void Send_color (uint8_t color)
- void Init_display ()
- void Write_pixel (uint8_t red, uint8_t green, uint8_t blue)
- void Set_cursor (uint16_t x, uint16_t y)
- void Create_partial_screen (uint16_t v_start, uint16_t v_end, uint8_t h_start, uint8_t h_end)

4.6.1 Detailed Description

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 Contain functions to control the screen.

4.6.2 Define Documentation

4.6.2.1 #define DISPLAY_CS 19

4.6.2.2 #define DISPLAY_RS 18

4.6.2.3 #define DISPLAY_WRB 20

4.6.2.4 #define LCD_HEIGHT 320

4.6.2.5 #define LCD_WIDTH 240

4.6.2.6 #define SIZE_LCD 76800

4.6.3 Function Documentation

4.6.3.1 void Create_partial_screen (uint16_t *v_start,* uint16_t *v_end,* uint8_t *h_start,* uint8_t *h_end*)

Creation of a window on the screen

create a window on the screen by precising a size

Parameters

v_start	largeur = v_start to v_end
v_end	
h_start	longueur = h_start to h_end
h_end	

4.6.3.2 void Index_out (uint8_t *idx*)

Permit to choose on which index we want to write on the screen options.

Parameters

idx	: Index address to access

4.6.3.3 void Init_display ( )

Screen initialisation

Set configuration: 8bits,262k color,CPU interface,Standby OFF,Set Gamma,Display on

4.6.3.4 void Init_ports_display ( )

Initialize all GPIO used with the screen.

4.6.3.5 void Parameter_out (uint16_t *param*)

Permit to change parameters on the index selected with the function Index_out.

Parameters

param	: Datas to write

4.6.3.6 void Send_color (uint8_t *color*)

Send a color to the touchscreen, this function need to be called 3 times to set all color on a pixel (R, G, B)

Parameters

color	Color sent

4.6.3.7 void Set_cursor (uint16_t *x,* uint16_t *y*)

Set cursor localisation

Parameters

x	value between 0 and 240
y	value between 0 and 320

4.6.3.8 void Write_pixel (uint8_t *red,* uint8_t *green,* uint8_t *blue*)

send RGB color on pixel

Parameters

red	value between 0 and 255
green	value between 0 and 255
blue	value between 0 and 255

4.7 ExtLab2.c File Reference

Contain functions to control the ExtLab2 card.

`#include "ExtLab2.h"` Include dependency graph for ExtLab2.c:

Functions

- void Init_Extlab2 ()

  Select the control bus on the ExtLab2 and restore the old
- void Select_control_bus ()

  Select the control bus on the ExtLab2 and restore the old
- void Select_display_bus ()

  Once the display bus selected all datas will directly sent to
- void Valide_datas_bus_to_extlab2 ()

  Switch the state on p2.8 to launch to validate the datas, but we also save the
- void Init_Rotate_button ()

  Active an interruption on the rotation of the button and

4.7.1 Detailed Description

Contain functions to control the ExtLab2 card.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.7.2 Function Documentation

4.7.2.1 void Init_Extlab2 ( )

Select the control bus on the ExtLab2 and restore the old

Select bus control on the ExtLab2 state of this one (with the variable save_data_bus_-values).

4.7.2.2 void Init_Rotate_button ( )

Active an interruption on the rotation of the button and

Initialization to use the rotate button configure the GPIO to use the rotate button

4.7.2.3 void Select_control_bus ( )

Select the control bus on the ExtLab2 and restore the old

Select bus control on the ExtLab2 state of this one (with the variable save_data_bus_-values).

4.7.2.4 void Select_display_bus ( )

Once the display bus selected all datas will directly sent to

Select the dispay bus on ExtLab2 the lcd screen

4.7.2.5 void Valide_datas_bus_to_extlab2 ( )

Switch the state on p2.8 to launch to validate the datas, but we also save the

Valide all datas put on the control bus datas (in case we switch to the display bus)

4.8 ExtLab2.h File Reference

Contain functions to control the ExtLab2 card.

`#include "LPC17xx.h" #include <cr_section_macros.h>` Include dependency graph for ExtLab2.h: This graph shows which files directly or indirectly include this file:

Functions

- void Select_display_bus ()

  Once the display bus selected all datas will directly sent to
- void Select_control_bus ()

Select the control bus on the ExtLab2 and restore the old

- void Valide_datas_bus_to_extlab2 ()

  Switch the state on p2.8 to launch to validate the datas, but we also save the

- void Init_Rotate_button ()

  Active an interruption on the rotation of the button and

- void Init_Extlab2 ()

  Select the control bus on the ExtLab2 and restore the old

4.8.1 Detailed Description

Contain functions to control the ExtLab2 card.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.8.2 Function Documentation

4.8.2.1 void Init_Extlab2 ( )

Select the control bus on the ExtLab2 and restore the old

Select bus control on the ExtLab2 state of this one (with the variable save_data_bus_-values).

4.8.2.2 void Init_Rotate_button ( )

Active an interruption on the rotation of the button and

Initialization to use the rotate button configure the GPIO to use the rotate button

4.8.2.3 void Select_control_bus ( )

Select the control bus on the ExtLab2 and restore the old

Select bus control on the ExtLab2 state of this one (with the variable save_data_bus_-values).

4.8.2.4 void Select_display_bus ( )

Once the display bus selected all datas will directly sent to

Select the dispay bus on ExtLab2 the lcd screen

4.8.2.5 void Valide_datas_bus_to_extlab2 ( )

Switch the state on p2.8 to launch to validate the datas, but we also save the

Valide all datas put on the control bus datas (in case we switch to the display bus)

4.9 main.c File Reference

This function is used to control the station with a touchscreen.

Defines

- #define MAX_TRAIN 100
- #define MAX_SPEED 1000
- #define INCREMENT_SPEED 50
- #define OPTION_SPEED 0
- #define OPTION_TRAIN_NUMBER 1
- #define TIME_ANTI_REBOUND 1
- #define SPI_RATE_TOUCHSCREEN 1500000

Functions

- char ∗ itoa (int val, int base)
- int atoi (char ∗str)
- void EINT3_IRQHandler (void)

  When someone press on the screen this interruption is called,.
- void TIMER0_IRQHandler ()

  This timer permit to ignore multiple pressures on the screen.
- int main (void)

Variables

- bool flag_interrupt = 0
- uint16_t lights [MAX_TRAIN]
- uint16_t speed_train [MAX_TRAIN]
- bool train_direction [MAX_TRAIN]
- int n_train = 1
- bool start_or_stop = 0

- uint8_t red [3] = { 255, 0, 0 }
- uint8_t green [3] = { 0, 255, 0 }
- uint8_t blue [3] = { 0, 0, 255 }
- uint8_t yellow [3] = { 255, 255, 0 }
- uint8_t color_button [3] = { 229, 208, 64 }
- uint8_t black [3] = { 0, 0, 0 }
- uint8_t option_selected = OPTION_TRAIN_NUMBER

4.9.1 Detailed Description

This function is used to control the station with a touchscreen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 You can select which train you want to control and it will display an image on the screen that correspond with the train controlled. To change the train select you need to use the rotation button. To change the speed you just press on the touchscreen the button with the text "Vitesse" and just increment with the rotate button. To turn the lights on or off, you just need to press on the button "Lumi" on the touchscreen.

4.9.2 Define Documentation

4.9.2.1 #define INCREMENT_SPEED 50

4.9.2.2 #define MAX_SPEED 1000

4.9.2.3 #define MAX_TRAIN 100

4.9.2.4 #define OPTION_SPEED 0

4.9.2.5 #define OPTION_TRAIN_NUMBER 1

4.9.2.6 #define SPI_RATE_TOUCHSCREEN 1500000

4.9.2.7 #define TIME_ANTI_REBOUND 1

4.9.3 Function Documentation

4.9.3.1 int atoi (char * str)

string to integer

Parameters

str	String to be converted to a integer

Returns

The value of the string converted

4.9.3.2 void EINT3_IRQHandler (void)

When someone press on the screen this interruption is called,.

Interruption generated by pressing on the touchscreen OR the rotate button we run a timer to ignore all rebounds. The "flag_interrupt", which valid the pressure, is only set after the end of timer. This flag is scrutinized in the main function. If we press the rotate button we just need to see if the other "edge" state to know which way we are turning.

4.9.3.3 char* itoa (int *val,* int *base*)

integer to string in the base desired

Parameters

val	Value to be converted to a string
base	Base on which we want the conversion

Returns

A pointer to the string that contain the conversion

4.9.3.4 int main (void)

Main program, initialize all devices and control the touchscreen pressures. If speed, light or direction if changed on the screen the program will send a frame through the UART that is connected on the XBEE. If someone change something on the station Marklin, all datas are actualized on station (ExtLab2) screen.

4.9.3.5 void TIMER0_IRQHandler ( )

This timer permit to ignore multiple pressures on the screen.

Interrupt generated by a timer when we press on the touchscreen because of rebounds.

4.9.4 Variable Documentation

4.9.4.1 uint8_t black[3] = { 0, 0, 0 }

4.9.4.2 uint8_t blue[3] = { 0, 0, 255 }

4.9.4.3 uint8_t color_button[3] = { 229, 208, 64 }

4.9.4.4 bool flag_interrupt = 0

4.9.4.5 uint8_t green[3] = { 0, 255, 0 }

4.9.4.6 uint16_t lights[MAX_TRAIN]

4.9.4.7 int n_train = 1

4.9.4.8 uint8_t option_selected = OPTION_TRAIN_NUMBER

4.9.4.9 uint8_t red[3] = { 255, 0, 0 }

4.9.4.10 uint16_t speed_train[MAX_TRAIN]

4.9.4.11 bool start_or_stop = 0

4.9.4.12 bool train_direction[MAX_TRAIN]

4.9.4.13 uint8_t yellow[3] = { 255, 255, 0 }

4.10 police.c File Reference

Write chosen letter on screen.

`#include "police.h"` Include dependency graph for police.c:

Functions

- void Write_char_with_background (char character, uint16_t x, uint16_t y, uint8_t *color_letter, uint8_t *color_background)
- void Write_char (char character, uint16_t x, uint16_t y, uint8_t *color_character)
- void Write_string_with_background (char *string, uint16_t x, uint16_t y, uint8_t *color_character, uint8_t *color_background)
- void Write_string (char *string, uint16_t x, uint16_t y, uint8_t *color_character)

Variables

- const unsigned char vga_fonts [ ]

4.10.1 Detailed Description

Write chosen letter on screen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.10.2 Function Documentation

4.10.2.1 void Write_char (char *character,* uint16_t *x,* uint16_t *y,* uint8_t * *color_character*)

Write chosen letter on screen

Parameters

character	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
*color_character	--> color of the char RGB

4.10.2.2 void Write_char_with_background (char *character,* uint16_t *x,* uint16_t *y,* uint8_t * *color_letter,* uint8_t * *color_background*)

Write chosen letter on screen

Parameters

character	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
*color_letter	--> color of the char RGB
*color_background	--> color of the background RGB

4.10.2.3 void Write_string (char * *string*, uint16_t *x*, uint16_t *y*, uint8_t * *color_character*)

Write chosen letter on screen

Parameters

string	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
**color_character*	--> color of the char RGB

4.10.2.4 void Write_string_with_background (char * *string*, uint16_t *x*, uint16_t *y*, uint8_t * *color_character*, uint8_t * *color_background*)

Write chosen letter on screen with background

Parameters

string	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
**color_character*	--> color of the char RGB
**color_background*	--> color of the background RGB

4.10.3 Variable Documentation

4.10.3.1 const unsigned char vga_fonts[ ]

4.11 police.h File Reference

content constants and header of functions used on police.c

`#include "LPC17xx.h"` `#include <cr_section_macros.h>` × `#include "Ecran.h"` Include dependency graph for police.h: This graph shows which files directly or indirectly include this file:

Defines

- #define LETTER_WIDTH 8
- #define LETTER_HEIGHT 22

Functions

- void Write_char (char character, uint16_t x, uint16_t y, uint8_t ∗color)
- void Write_string (char ∗string, uint16_t x, uint16_t y, uint8_t ∗color_character)
- void Write_string_with_background (char ∗string, uint16_t x, uint16_t y, uint8_t ∗color_character, uint8_t ∗color_background)
- void Write_char_with_background (char character, uint16_t x, uint16_t y, uint8_t ∗color_letter, uint8_t ∗color_background)
- void Create_button (char ∗string, uint16_t x, uint16_t y, uint8_t hauteur, uint8_t largeur)

4.11.1 Detailed Description

content constants and header of functions used on police.c

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.11.2 Define Documentation

4.11.2.1 #define LETTER_HEIGHT 22

4.11.2.2 #define LETTER_WIDTH 8

4.11.3 Function Documentation

4.11.3.1 void Create_button (char ∗ *string,* uint16_t *x,* uint16_t *y,* uint8_t *hauteur,* uint8_t *largeur*)

4.11.3.2 void Write_char (char *character,* uint16_t *x,* uint16_t *y,* uint8_t ∗ *color_character*)

Write chosen letter on screen

Parameters

character	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
∗*color_character*	--> color of the char RGB

4.11.3.3 void Write_char_with_background (char *character,* uint16_t *x,* uint16_t *y,* uint8_t * *color_letter,* uint8_t * *color_background*)

Write chosen letter on screen

Parameters

character	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
*color_letter	--> color of the char RGB
*color_background	--> color of the background RGB

4.11.3.4 void Write_string (char * *string,* uint16_t *x,* uint16_t *y,* uint8_t * *color_character*)

Write chosen letter on screen

Parameters

string	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
*color_character	--> color of the char RGB

4.11.3.5 void Write_string_with_background (char * *string,* uint16_t *x,* uint16_t *y,* uint8_t * *color_character,* uint8_t * *color_background*)

Write chosen letter on screen with background

Parameters

string	
x	--> position x on the screen (0 - 239)
y	--> position y on the screen (0 - 319)
*color_character	--> color of the char RGB
*color_background	--> color of the background RGB

4.12 SD.c File Reference

`#include "SD.h"` Include dependency graph for SD.c:

Functions

- void Clear_pin ()
- void Set_pin ()
- void Read_SD_one_block (uint32_t n_block)
- void Read_SD_multi_block (uint32_t n_block_depart, uint32_t n_block_arrive)
- void Write_SD ()
- void ChipSetSelect ()
- void init_SD ()

Variables

- int x
- int i
- int y
- int data
- int k
- int l = 0
- uint8_t rep
- int reponse [512]
- uint8_t Tab [512]

4.12.1 Function Documentation

4.12.1.1 void ChipSetSelect ( )

4.12.1.2 void Clear_pin ( )

4.12.1.3 void init_SD ( )

4.12.1.4 void Read_SD_multi_block (uint32_t *n_block_depart*, uint32_t *n_block_arrive*)

4.12.1.5 void Read_SD_one_block (uint32_t *n_block*)

4.12.1.6 void Set_pin ( )

4.12.1.7 void Write_SD ( )

4.12.2 Variable Documentation

4.12.2.1 int data

4.12.2.2 int i

4.12.2.3 int k

4.12.2.4 int l = 0

4.12.2.5 uint8_t rep

4.12.2.6 int reponse[512]

4.12.2.7 uint8_t Tab[512]

4.12.2.8 int x

4.12.2.9 int y

4.13 SD.h File Reference

```
#include "LPC17xx.h"      #include <cr_section_macros.h> ×
#include "SPI.h"  #include "ExtLab2.h"  #include "Ecran.h"
```

Include dependency graph for SD.h: This graph shows which files directly or indirectly include this file:

Functions

- void init_SD ()
- void Read_SD_one_block ()
- void Read_SD_multi_block (uint32_t n_block_depart, uint32_t n_block_arrive)
- void Clear_pin ()
- void Set_pin ()
- void Write_SD ()
- void Read_SD_multiBLocks ()

4.13.1 Function Documentation

4.13.1.1 void Clear_pin ( )

4.13.1.2 void init_SD ( )

4.13.1.3 void Read_SD_multi_block (uint32_t *n_block_depart,* uint32_t *n_block_arrive*)

4.13.1.4 void Read_SD_multiBLocks ( )

4.13.1.5 void Read_SD_one_block ( )

4.13.1.6 void Set_pin ( )

4.13.1.7 void Write_SD ( )

4.14 SendUARTFormat.c File Reference

Contain function to send frames on the UART with a defined.

`#include "SendUARTFormat.h"` Include dependency graph for SendUARTFormat.c:

Functions

- void send_speed (uint8_t n_train, uint16_t speed_train)
- void send_direction (uint8_t n_train, bool direction)
- void send_lights (uint8_t n_train, bool state)

4.14.1 Detailed Description

Contain function to send frames on the UART with a defined.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 syntax to control the train.

4.14.2 Function Documentation

4.14.2.1 void send_direction (uint8_t *n_train*, bool *direction*)

Send the direction on the UART to control the chosen train.

Parameters

n_train	Train number
direction	The direction sent to the train

4.14.2.2 void send_lights (uint8_t *n_train*, bool *state*)

Send the state of the lights on the UART to control the chosen train.

Parameters

n_train	Train number
state	Turn on (true) or turn off (false) on the chosen train.

4.14.2.3 void send_speed (uint8_t *n_train*, uint16_t *speed_train*)

Send the speed on the UART to control the chosen train.

Parameters

n_train	Train number
speed_train	The speed sent to the train

4.15 SendUARTFormat.h File Reference

Contain function to send frames on the UART with a defined.

`#include "uart.h"` `#include "stdbool.h"` Include dependency graph for SendUARTFormat.h: This graph shows which files directly or indirectly include this file:

4.15.1 Detailed Description

Contain function to send frames on the UART with a defined.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014 syntax to control the train.

4.16 SPI.c File Reference

Contain all function to communication with the SPI.

`#include "SPI.h"` Include dependency graph for SPI.c:

Functions

- void Init_SPI_master_mode (uint8_t S_CPHA, uint8_t S_CPOL, uint32_t SPI_rate, uint8_t _16_or_8bits_com)

- void Change_Frequency_SPI (uint32_t SPI_rate)
- void Write_only_SPI_8bits (uint8_t data)
- uint8_t Write_Read_SPI_8bits (uint8_t data)

4.16.1 Detailed Description

Contain all function to communication with the SPI.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.16.2 Function Documentation

4.16.2.1 void Change_Frequency_SPI (uint32_t *SPI_rate*)

Initialization of master mode on SPI

Parameters

SPI_rate	--> frequency SPI

4.16.2.2 void Init_SPI_master_mode (uint8_t *S_CPHA*, uint8_t *S_CPOL*, uint32_t *SPI_rate*, uint8_t *_16_or_8bits_com*)

Initialization of master mode on SPI

Parameters

S_CPHA	--> clock phase control
S_CPOL	--> rising/failing edge
SPI_rate	--> rate of SPI
_16_or_-8bits_com	--> select 16 bits or 8 bits mode

4.16.2.3 void Write_only_SPI_8bits (uint8_t *data*)

send data to SPI without return value

Parameters

datas	--> data to send on SPI

4.16.2.4 uint8_t Write_Read_SPI_8bits (uint8_t *data*)

send data on SPI bus and read the data received during the

Parameters

data	--> data to send on SPI

Returns

data received on SPI

4.17 SPI.h File Reference

Contain all function to communication with the SPI.

`#include "LPC17xx.h" #include <cr_section_macros.h>` Include dependency graph for SPI.h: This graph shows which files directly or indirectly include this file:

Defines

- #define BIT_ENABLE 2
- #define CPHA 3
- #define CPOL 4
- #define MODE_SELECT 5
- #define SPIF 7

Functions

- void Write_only_SPI_8bits (uint8_t data)
- uint8_t Write_Read_SPI_8bits (uint8_t data)
- void Init_SPI_master_mode (uint8_t S_CPHA, uint8_t S_CPOL, uint32_t SPI_rate, uint8_t _16_or_8bits_com)
- void Change_Frequency_SPI (uint32_t SPI_rate)

4.17.1 Detailed Description

Contain all function to communication with the SPI.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.17.2 Define Documentation

4.17.2.1 #define BIT_ENABLE 2

4.17.2.2 #define CPHA 3

4.17.2.3 #define CPOL 4

4.17.2.4 #define MODE_SELECT 5

4.17.2.5 #define SPIF 7

4.17.3 Function Documentation

4.17.3.1 void Change_Frequency_SPI (uint32_t *SPI_rate*)

Initialization of master mode on SPI

Parameters

SPI_rate	--> frequency SPI

4.17.3.2 void Init_SPI_master_mode (uint8_t *S_CPHA*, uint8_t *S_CPOL*, uint32_t *SPI_rate*, uint8_t *_16_or_8bits_com*)

Initialization of master mode on SPI

Parameters

S_CPHA	--> clock phase control
S_CPOL	--> rising/failing edge
SPI_rate	--> rate of SPI

_16_or_-8bits_com	--> select 16 bits or 8 bits mode

4.17.3.3 void Write_only_SPI_8bits (uint8_t *data*)

send data to SPI without return value

Parameters

datas	--> data to send on SPI

4.17.3.4 uint8_t Write_Read_SPI_8bits (uint8_t *data*)

send data on SPI bus and read the data received during the

Parameters

data	--> data to send on SPI

Returns

data received on SPI

4.18 Touchscreen.c File Reference

Contain all function configure and read the values from the touchscreen.

`#include "Touchscreen.h"` Include dependency graph for Touchscreen.c:

Functions

- void Init_touchscreen ()
- uint16_t Read_x_12bits ()
- uint16_t Read_y_12bits ()
- void Read_x_and_y_12bits (uint16_t *x, uint16_t *y)

4.18.1 Detailed Description

Contain all function configure and read the values from the touchscreen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.18.2 Function Documentation

4.18.2.1 void Init_touchscreen ( )

Initialization on touchscreen, set an interruption when we press touchscreen. This interruption is linked to EINT3.

4.18.2.2 uint16_t Read_x_12bits ( )

Read the x value from the touchscreen

Returns

x value coded between 0 and 4096

4.18.2.3 void Read_x_and_y_12bits (uint16_t * *x*, uint16_t * *y*)

Read the x and y values from the touchscreen

Returns

x and y values coded between 0 and 4096

4.18.2.4 uint16_t Read_y_12bits ( )

Read the y value from the touchscreen

Returns

y value coded between 0 and 4096

4.19 Touchscreen.h File Reference

Contain all function configure and read the values from the touchscreen.

`#include "LPC17xx.h"` `#include <cr_section_macros.h>` × `#include "SPI.h"` `#include "Ecran.h"` Include dependency graph for Touchscreen.h: This graph shows which files directly or indirectly include this file:

Defines

- #define IRQ_Touchscreen 0
- #define ExtLab2_IRQ 10
- #define CS_touchscreen 8

Functions

- void Init_touchscreen ()
- uint16_t Read_y_12bits ()
- uint16_t Read_x_12bits ()
- void Read_x_and_y_12bits (uint16_t *x, uint16_t *y)

4.19.1 Detailed Description

Contain all function configure and read the values from the touchscreen.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.19.2 Define Documentation

4.19.2.1 #define CS_touchscreen 8

4.19.2.2 #define ExtLab2_IRQ 10

4.19.2.3 #define IRQ_Touchscreen 0

4.19.3 Function Documentation

4.19.3.1 void Init_touchscreen ( )

Initialization on touchscreen, set an interruption when we press touchscreen. This interruption is linked to EINT3.

4.19.3.2 uint16_t Read_x_12bits ( )

Read the x value from the touchscreen

Returns

x value coded between 0 and 4096

4.19.3.3 void Read_x_and_y_12bits (uint16_t * *x*, uint16_t * *y*)

Read the x and y values from the touchscreen

Returns

x and y values coded between 0 and 4096

4.19.3.4 uint16_t Read_y_12bits ( )

Read the y value from the touchscreen

Returns

y value coded between 0 and 4096

4.20 uart.c File Reference

Contain all functions to initialize, write and read on UART 0 and 3.

`#include "LPC17xx.h"` `#include "uart.h"` Include dependency graph for uart.c:

Functions

- void uart0_init (uint32_t baudrate)
- void uart3_init (uint32_t baudrate)
- void uart0_send (char *data, uint32_t length)
- uint32_t uart0_read (char *data, uint32_t length)
- uint32_t uart0_read_one_char (char *ch)
- void uart3_send (char *data, uint32_t length)
- uint32_t uart3_read (char *data, uint32_t length)
- uint32_t uart3_read_one_char (char *ch)

4.20.1 Detailed Description

Contain all functions to initialize, write and read on UART 0 and 3.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.20.2 Function Documentation

4.20.2.1 void uart0_init (uint32_t *baudrate*)

Initialize UART0 port, setup pin select, clock, parity, stop bits, FIFO, etc.

Parameters

baudrate	UART0 baudrate [bit/s]

4.20.2.2 uint32_t uart0_read (char * *data,* uint32_t *length*)

Read data from UART0

Parameters

data	Pointer that store the data read from UART 0
length	Number of bytes to read

4.20.2.3 uint32_t uart0_read_one_char (char * *ch*)

Read one byte from UART0

Parameters

ch	Pointer that store the byte read from UART 0

4.20.2.4 void uart0_send (char * data, uint32_t length)

Send data on UART0

Parameters

data	Pointer on the datas to be sent
baudrate	Number of bytes to send

4.20.2.5 void uart3_init (uint32_t baudrate)

Initialize UART3 port, setup pin select, clock, parity, stop bits, FIFO, etc.

Parameters

baudrate	UART3 baudrate [bit/s]

4.20.2.6 uint32_t uart3_read (char * data, uint32_t length)

Read data from UART3

Parameters

data	Pointer that store the data read from UART 3
length	Number of bytes to read

4.20.2.7 uint32_t uart3_read_one_char (char * ch)

Read one byte from UART3

Parameters

ch	Pointer that store the byte read from UART 3

4.20.2.8 void uart3_send (char * data, uint32_t length)

Send data on UART3

Parameters

data	Pointer on the datas to be sent
baudrate	Number of bytes to send

4.21 uart.h File Reference

Contain all functions to initialize, write and read on UART 0 and 3.

`#include <stdint.h>` Include dependency graph for uart.h: This graph shows which files directly or indirectly include this file:

Defines

- #define LSR_RDR 0x01
- #define LSR_OE 0x02
- #define LSR_PE 0x04
- #define LSR_FE 0x08
- #define LSR_BI 0x10
- #define LSR_THRE 0x20
- #define LSR_TEMT 0x40
- #define LSR_RXFE 0x80

Functions

- void uart0_init (uint32_t baudrate)
- void uart0_send (char ∗data, uint32_t length)
- uint32_t uart0_read (char ∗data, uint32_t length)
- uint32_t uart0_read_one_char (char ∗ch)
- void uart3_init (uint32_t baudrate)
- void uart3_send (char ∗data, uint32_t length)
- uint32_t uart3_read (char ∗data, uint32_t length)
- uint32_t uart3_read_one_char (char ∗ch)

4.21.1 Detailed Description

Contain all functions to initialize, write and read on UART 0 and 3.

Author

Da Silva Andrade David, Antoine Berger, Dos Santos Rafael

Version

1.0

Date

19 June 2014

4.21.2 Define Documentation

4.21.2.1 #define LSR_BI 0x10

4.21.2.2 #define LSR_FE 0x08

4.21.2.3 #define LSR_OE 0x02

4.21.2.4 #define LSR_PE 0x04

4.21.2.5 #define LSR_RDR 0x01

4.21.2.6 #define LSR_RXFE 0x80

4.21.2.7 #define LSR_TEMT 0x40

4.21.2.8 #define LSR_THRE 0x20

4.21.3 Function Documentation

4.21.3.1 void uart0_init (uint32_t *baudrate*)

Initialize UART0 port, setup pin select, clock, parity, stop bits, FIFO, etc.

Parameters

baudrate	UART0 baudrate [bit/s]

4.21.3.2 uint32_t uart0_read (char * *data,* uint32_t *length*)

Read data from UART0

Parameters

data	Pointer that store the data read from UART 0
length	Number of bytes to read

4.21.3.3 uint32_t uart0_read_one_char (char * *ch*)

Read one byte from UART0

Parameters

ch	Pointer that store the byte read from UART 0

4.21.3.4 void uart0_send (char * data, uint32_t length)

Send data on UART0

Parameters

data	Pointer on the datas to be sent
baudrate	Number of bytes to send

4.21.3.5 void uart3_init (uint32_t baudrate)

Initialize UART3 port, setup pin select, clock, parity, stop bits, FIFO, etc.

Parameters

baudrate	UART3 baudrate [bit/s]

4.21.3.6 uint32_t uart3_read (char * data, uint32_t length)

Read data from UART3

Parameters

data	Pointer that store the data read from UART 3
length	Number of bytes to read

4.21.3.7 uint32_t uart3_read_one_char (char * ch)

Read one byte from UART3

Parameters

ch	Pointer that store the byte read from UART 3

4.21.3.8 void uart3_send (char * data, uint32_t length)

Send data on UART3

Parameters

data	Pointer on the datas to be sent
baudrate	Number of bytes to send