DMX512 Controller Receiver

Generated by Doxygen 1.8.15

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

satej_matthew.c
File containing everything for the DMX Controller Receiver Project.
For CSE 4342: Embedded II Spring 2019
Instructor: Dr. Jason Losh
3
tm4c123gh6pm_startup_ccs.c
Startup File for Project . . . . **??**

Chapter 2

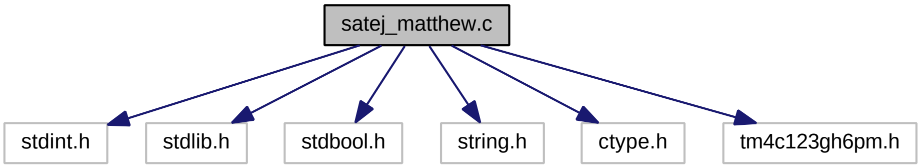
File Documentation

2.1 satej_matthew.c File Reference

File containing everything for the DMX Controller Receiver Project.
For CSE 4342: Embedded II Spring 2019
Instructor: Dr. Jason Losh

```
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
#include "tm4c123gh6pm.h"
```

Include dependency graph for satej_matthew.c:



Macros

- #define RED_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 1*4)))
- #define GREEN_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 3*4)))
- #define BLUE_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 2*4)))
- #define PUSH_BUTTON (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 4*4)))
- #define PUSH_BUTTON2 (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 0*4)))
- #define GREEN_LED_MASK 8
- #define RED_LED_MASK 2
- #define BLUE_LED_MASK 4
- #define PUSH_BUTTON_MASK 16
- #define PUSH_BUTTON2_MASK 1
- #define delay4Cycles() __asm(" NOP\n NOP\n NOP\n NOP")
- #define delay1Cycle() __asm(" NOP\n")
- #define delay6Cycles() __asm(" NOP\n NOP\n NOP\n NOP\n NOP\n NOP\n")

Functions

- void animationRamp ()

  Function to enable ramping animation.
- void clearStr ()

  Function to clear command, arg1, and arg2 arrays.
- char getcUart0 ()

  Blocking function that returns with serial data once the buffer is not empty.
- void getModeEE ()

  Function to get the launchpad mode from EEPROM.
- char ∗ intToChar (uint16_t x)

  Function to convert integer to character for UART0.
- bool isLetter (char c)

  Function to check if character is letter.
- bool isNumber (char c)

  Function to check if character is number.
- uint8_t main ()

  Runs everything.
- void printCommandList ()

  Function to print available commands to user.
- void putcUart1 (uint8_t i)

  Function to send characters to UART0.
- void Uart0Isr ()

  Function to handle UART0 interrupts.
- void waitMicrosecond (uint32_t us)

  Function to wait for specified microseconds.
- void wooone ()

  Function to set all DMX values to 255.
- void putsUart0 (char ∗str)

  Blocking function that writes a string when the UART buffer is not full.
- void changeTimer1Value (uint32_t us)

  Function to change load value of Timer1.
- void initHw ()

  Function to initialize all required hardware functions.
- void Uart1Isr ()

  Function to Handle Interrupts from UART1.
- void Timer2ISR ()

  Function to Handle Interrupts from Timer2.
- void Timer1ISR ()

  Function to handle TIMER1 interrupts.
- void putcUart0 (char c)

  Blocking function that writes a serial character when the UART buffer is not full.
- void EEWRITE (uint16_t B, uint16_t offSet, uint16_t val)

  Function to write to EEPROM to set address.
- void clearDMX ()

  Function to clear DMX data bins.
- uint8_t parseCommand ()

  Function to parse commands from UART0 and execute functions or set flags.
- void sweepServo ()

  Function to sweep servo.

Variables

- char command [20]
- char arg1 [20]
- char arg2 [20]
- int8_t enteringField = 0
- int8_t pos = 0
- uint16_t maxAddress = 512
- uint8_t continuous = 0
- uint16_t DMXMode = 0
- uint16_t deviceModeAddress = 0
- uint8_t prevRX = 0
- uint8_t rxError = 0
- uint16_t rxState = 0
- float seconds = 0
- int upR
- int upG
- int upB
- int goR
- int goG
- int goB
- float secondsTrigger = 0.0
- uint16_t dimStart = 0
- uint16_t dimEnd = 0
- float dimValue = 0
- uint8_t woo = 0
- int servoDir = 0
- char ch [3]
- uint8_t vall = 8
- uint8_t incr = 1
- uint16_t program
- uint16_t Address
- uint16_t opMode
- uint16_t setval
- uint8_t mode = 0
- uint8_t dmxData [512]
- uint8_t RGBMode = 0

2.1.1 Detailed Description

File containing everything for the DMX Controller Receiver Project.
For CSE 4342: Embedded II Spring 2019
Instructor: Dr. Jason Losh

Author

Satej Mhatre, Matthew Hilliard

Date

1 May 2019

Hardware Target:

Target Platform: EK-TM4C123GXL Evaluation Board
Target uC: TM4C123GH6PM
System Clock: 40 MHz

Hardware configuration:

Red LED:
PF1 drives an NPN transistor that powers the red LED
Blue LED:
PF2 drives an NPN transistor that powers the green LED
Green LED:
PF3 drives an NPN transistor that powers the green LED
UART Interface:
U0TX (PA1) and U0RX (PA0) are connected to the 2nd controller
U1TX (PA1) and U1RX (PA0) are used for DMX Data Transmit and Receive
Other Interface:
PD0, PD1, PD2, PD3 is connected to a mux that reads the value from a DIP switch
PF1, PF2, PF3 are also configured as PWM outputs to control servos and LEDs on-board.
To Do:
PD6, PD7 will be connected to a ESP8266-01 that will serve a webpage for UART communication so that launchpad can be controlled without physically using a USB cable.
The USB on the 2nd controller enumerates to an ICDI interface and a virtual COM port
Configured to 115,200 baud, 8N1

Definition in file satej_matthew.c.

2.1.2 Macro Definition Documentation

2.1.2.1 BLUE_LED

```
#define BLUE_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 2*4)))
```

Bit banding for PORTF2 Blue LED

Definition at line 51 of file satej_matthew.c.

2.1.2.2 BLUE_LED_MASK

```
#define BLUE_LED_MASK 4
```

GPIO PORTF Blue LED Mask

Definition at line 66 of file satej_matthew.c.

2.1.2.3 delay1Cycle

```
#define delay1Cycle( ) __asm(" NOP\n")
```

Delaying for 1 cycle

Definition at line 77 of file satej_matthew.c.

2.1.2.4 delay4Cycles

```
#define delay4Cycles( ) __asm(" NOP\n NOP\n NOP\n NOP")
```

Delaying for 4 cycles

Definition at line 75 of file satej_matthew.c.

2.1.2.5 delay6Cycles

```
#define delay6Cycles( ) __asm(" NOP\n NOP\n NOP\n NOP\n NOP\n NOP\n")
```

Delaying for 6 cycles

Definition at line 79 of file satej_matthew.c.

2.1.2.6 GREEN_LED

```
#define GREEN_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 3*4)))
```

Bit banding for PORTF3 GREEN LED

Definition at line 48 of file satej_matthew.c.

2.1.2.7 GREEN_LED_MASK

```
#define GREEN_LED_MASK 8
```

GPIO PORTF Green LED Mask

Definition at line 60 of file satej_matthew.c.

2.1.2.8 PUSH_BUTTON

```
#define PUSH_BUTTON (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 4*4)))
```

Bit banding for PORTF4 PushButton 1

Definition at line 54 of file satej_matthew.c.

2.1.2.9 PUSH_BUTTON2

```
#define PUSH_BUTTON2 (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 0*4)))
```

Bit banding for PORTF0 PushButton 0

Definition at line 57 of file satej_matthew.c.

2.1.2.10 PUSH_BUTTON2_MASK

```
#define PUSH_BUTTON2_MASK 1
```

GPIO PORTF Push Button 2 Mask

Definition at line 72 of file satej_matthew.c.

2.1.2.11 PUSH_BUTTON_MASK

```
#define PUSH_BUTTON_MASK 16
```

GPIO PORTF Push Button 1 Mask

Definition at line 69 of file satej_matthew.c.

2.1.2.12 RED_LED

```
#define RED_LED (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 1*4)))
```

Bit banding for PORTF1 Red LED

Definition at line 45 of file satej_matthew.c.