Tinkerpal Docs Documentation

Release latest

April 05, 2016

Contents

Contents:

Objects and methods detailed below

CHAPTER 1

Global

1.1 eval

Evaluate the given String

1.1.1 Example

```
var two = eval('1+1');
```

Parameter Name	Description
str	String to be evaluated

1.2 toInteger

Convert the given object to an integer value

1.2.1 Example

```
var n = toInteger('5') + 1;
```

Parameter Name	Description
object	Object to be converted

1.3 isNaN

Tests if the object is NaN

1.3.1 Example

```
isNaN(1); // false
isNaN('abc'); // true
```

Parameter Name	Description
number	Object to be tested for NaNity

1.4 meminfo

Prints platform dependent memory information

1.4.1 Example

```
meminfo()
```

1.5 describe

Print object description to the console

1.5.1 Example

```
describe(describe);
```

Parameter Name	Description
object	Object to describe

1.6 compile

Compiles a function for faster execution

1.6.1 Example

```
var f = compile(function() { return 1 + 1; });
```

Parameter Name	Description
function	function_to_be_compiled

1.7 objGraph

Create a DOT graph of allocated objects

1.7.1 Example

```
objGraph()
```

CHAPTER 2

File System

2.1 readFileSync

Synchronously reads the entire contents of a file

2.1.1 Example

```
var s = fs.readFileSync('FAT/file.txt');
```

Parameter Name	Description
path	File Path

2.2 writeFileSync

Synchronously writes the entire contents of a file

2.2.1 Example

```
fs.writeFileSync('FAT/file.txt', 'hello world!');
```

Parameter Name	Description
path	File Path
data	String to be written to file

2.3 readdirSync

Synchronously reads the directory listing of a path

2.3.1 Example

```
var s = fs.readdirSync('FAT/');
```

Parameter Name	Description
path	Directory Path

CHAPTER 3

Array

3.1 Array (constructor)

Array Constructor

3.1.1 Example

```
var a = new Array(5);
var b = new Array(1, 2, 3);
```

Para mete r Name	Desc ript ion
Leng th or [ite m1, item 2, ..]	Numb er of entr ies to Allo cate or comm a sepa rate d list of item s to push to the new arra y

3.2 push

The arguments are appended to the end of the array, in the order in which they appear

3.2.1 Example

```
var a = [];
a.push(1);
```

Parameter Name	Description
item1 [, item2 [, ...]]	object[s] to be added

3.3 pop

The last element of the array is removed from the array and returned

3.3.1 Example

```
var a = [2, 1];
var one = a.pop();
```

3.4 forEach

Calls cb once for each element present in the array, in ascending order

3.4.1 Example

```
var a = [1, 2, 3];
a.forEach(function(value, k, obj) { console.log(value + ' [' + k + '] ' + ' @ ' + obj); });
```

Para mete r Name	Desc ript ion
cb	func tion that acce pts thre e argu ment s:Va lue of the elem ent, Inde x of the elem ent, The obje ct bein g trav erse d
this	[opt iona l] if prov ided , used as the 'thi s' valu e for each invo cati on of 'cb' . If not prov ided , 'und efin ed' is used

3.5 indexOf

Searches for searchElement in the elements of the array

3.5.1 Example

```
var a = [1, 2, 3];
var one = a.indexOf(2);
```

Parameter Name	Description
searchElement	Object to search for
fromIndex	[optional] start search index

3.6 join

Joins the string coversions of the elements of the array separated by occurrences of the separator

3.6.1 Example

```
var a = [1, 2, 3];
debug.assert(a.join('-'), '1-2-3');
```

Parameter Name	Description
separator	[optional] string. If not provided, a single comma is used

3.7 map

Calls cb once for each element present in the array, in ascending order, and constructs a new Array from the results

3.7.1 Example

```
var a = [1, 2, 3];
var b = a.map(function(value, k, obj) { return value + 1 });
debug.assert(b, [2, 3, 4]);
```

Para mete r Name	Desc ript ion
cb	func tion that acce pts thre e argu ment s:Va lue of the elem ent, Inde x of the elem ent, The obje ct bein g trav erse d
this	[opt iona l] if prov ided , used as the 'thi s' valu e for each invo cati on of 'cb' . If not prov ided , 'und efin ed' is used

3.8 slice

Creates a new array with elements from the specified array starting from index 'start' up to index 'end'

3.8.1 Example

```
[1, 2, 3].slice(1); // [2, 3]
```

Parameter Name	Description
start	Element index to start from, may be undefined
end	Last element index (not inclusive), may be undefined

3.9 sort

Sorts the array elements

3.9.1 Example

```
[3, 1, 2].sort(); // [1, 2, 3]
```

Para mete r Name	Desc ript ion
comp aref n	Func tion rece ivin g (x, y) and retu rns a nega tive valu e if x < y, zero if x = y or a posi tive valu e if x > y

3.10 filter

Calls cb once for each element present in the array, in ascending order, and constructs a new Array from any element for which cb returned 'true'

3.10.1 Example

```
var a = [1, 2, 3];
var b = a.filter(function(value) { return value > 1 });
b; // [2, 3]
```

Para mete r Name	Desc ript ion
cb	func tion that acce pts thre e argu ment s:Va lue of the elem ent, Inde x of the elem ent, The obje ct bein g trav erse d
this	[opt iona l] if prov ided , used as the 'thi s' valu e for each invo cati on of 'cb' . If not prov ided , 'und efin ed' is used

3.11 concat

Concatanates a given array with a list of items. If an item is an array itself, its members are used

3.11.1 Example

```
var a = [1, 2, 3];
var b = a.concat([4, 5, 6], 7); // [1, 2, 3, 4, 5, 6, 7]
```

Parameter Name	Description
[item1 [, item2 [, item3]]]	optional list of items to concatanate

CHAPTER 4

String

4.1 String (constructor)

String Constructor

4.1.1 Example

```
var s = new String('hello');
```

Parameter Name	Description
String	String

4.2 split

Breaks a string into substrings based on occurences of 'separator'

4.2.1 Example

```
var s = '1|2|3';
var a = s.split('|');
debug.assert(a, [ '1', '2', '3' ]);
```

Parameter Name	Description
separator	Delimiter

4.3 indexOf

Search for occurences of 'searchString' in a given string

4.3.1 Example

```
var s = 'looking for me';
var i = s.indexOf('for');
debug.assert(i, 8);
```

Parameter Name	Description
searchString	String to search for

4.4 substring

Creates a new string based on a subset of a given string

4.4.1 Example

```
var s = 'a big string';
var big = s.substring(2, 5);
```

Parameter Name	Description
start	Character position to start from
end	Character position to end with (not including

4.5 charAt

Creates a new string containing the character at a position

4.5.1 Example

```
var s = 'a string';
var a = s.charAt(0);
```

Parameter Name	Description
pos	Position of character

4.6 charCodeAt

Get the ASCII value of the character at a position

4.6.1 Example

```
var s = 'a string';
var ninty_seven = s.charCodeAt(0);
```

Parameter Name	Description
pos	Position of character

4.7 toLowerCase

Convert a string to lower case characters

4.7.1 Example

```
var s = 'Hello World';
s.toLowerCase(); // 'hello world'
```

4.8 toUpperCase

Convert a string to upper case characters

4.8.1 Example

```
var s = 'Hello World';
s.toUpperCase(); // 'HELLO WORLD'
```

CHAPTER 5

Function

5.1 Function (constructor)

Function Constructor

5.1.1 Example

```
var f = new Function('a', 'b', 'return a+b');
console.log('1+2=' + f(1, 2));
```

Parameter Name	Description
[arg1, [arg2, [..]]]	Formal Parameter Names
Body	Function Body

5.2 call

Invoke function call

5.2.1 Example

```
function fun() { this.prop = 'prop' }
var obj = {};
fun.call(obj);
obj; // { prop : 'prop' }
```

Parameter Name	Description
thisArg	The 'this' value on function invocation
arg1, arg2, ...	[optional] arguments to pass to function

5.3 apply

Invoke function call

5.3.1 Example

```
function fun(a, b) { return a + b; }
fun.apply(undefined, [1, 2]); // 3
```

Parameter Name	Description
thisArg	The 'this' value on function invocation
argArray	[optional] Array of arguments to pass to function

5.4 bind

Returns a new Function object based on the current function object, but the 'this' value is bound to thisArg

5.4.1 Example

```
function fun() { return this.prop; }
var f = fun.bind({ prop : 1});
f(); // 1
```

Parameter Name	Description
thisArg	The 'this' value on function invokation

CHAPTER 6

ArrayBuffer

6.1 ArrayBuffer (constructor)

ArrayBuffer Constructor

6.1.1 Example

```
var a = new ArrayBuffer(5);
```

Parameter Name	Description
Size	Number of bytes in Array Buffer

CHAPTER 7

ArrayBufferView

7.1 Int8Array (constructor)

Int8Array Constructor

7.1.1 Example

```
var a = new Int8Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.2 Uint8Array (constructor)

Uint8Array Constructor

7.2.1 Example

```
var a = new Uint8Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.3 Int16Array (constructor)

Int16Array Constructor

7.3.1 Example

```
var a = new Int16Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.4 Uint16Array (constructor)

Uint16Array Constructor

7.4.1 Example

```
var a = new Uint16Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.5 Int32Array (constructor)

Int32Array Constructor

7.5.1 Example

```
var a = new Int32Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.6 Uint32Array (constructor)

Uint32Array Constructor

7.6.1 Example

```
var a = new Uint32Array(5);
```

Para mete r Name	Desc ript ion
Arra y/Ar rayB uffe r/Si ze	Arra y, Arra yBuf fer or Numb er of Elem ents in a newl y crea ted Arra yBuf fer

7.7 subarray

Return a partial typed array based on the typed arrayIf end is unspecified, the subarray contains all elements from begin to the end of the TypedArray. If either begin or end are negative, they are calculated from the end of the array

7.7.1 Example

```
var a = new Int8Array(16);
var b = a.subarray(1, 5);
```

Parameter Name	Description
begin	Start offset (inclusive)
end	(Optional) End offset (exclusive)

CHAPTER 8

Object

8.1 Object (constructor)

Object Constructor

8.1.1 Example

```
var o = new Object(5);
```

8.2 toString

The object is converted to a string

8.2.1 Example

```
var a = 1;
debug.assert(a.toString(), '1');
```

Parameter Name	Description
radix	(optional) radix to use in conversion

8.3 on

Adds a listener for the specified event

8.3.1 Example

```
a.on('data', function() { console.log('data!'); });
```

Parameter Name	Description
event	event to listen on
cb	callback function called on event

8.4 emit

Execute each of the listeners on the event

8.4.1 Example

```
a.emit('data');
```

Parameter Name	Description
event	event to listen on

8.5 removeAllListeners

removes listeners on a specified event, or all events

8.5.1 Example

```
a.removeAllListeners('data');
```

Parameter Name	Description
event	(optional) event to stop listening on

8.6 listeners

Returns an array with listeners for the specified event

8.6.1 Example

```
a.listeners('data');
```

Parameter Name	Description
event	event for fetching listeners

CHAPTER 9

debug

9.1 assert

Panic on mismatch between two objects

9.1.1 Example

```
debug.assert(1, 1);
```

Parameter Name	Description
object1	Object for comparison
object2	Object for comparison

9.2 assert_cond

Panic if object is false

9.2.1 Example

```
debug.assert_cond(1 == 1);
```

Parameter Name	Description
object	Object to test

9.3 assert_exception

Panic if calling cb() does not raise an exception

9.3.1 Example

```
debug.assert_exception(function() { throw 'error'; });
```

Parameter Name	Description
cb	function to test

9.4 dump_env

Dump global environment information to the console

9.4.1 Example

```
debug.dump_env();
```

CHAPTER 10

Math

10.1 sin

Compute the sine of an angle

10.1.1 Example

```
var zero = Math.sin(Pi);
```

Parameter Name	Description
angle	angle in radians

10.2 asin

Compute the arc sine of a number

10.2.1 Example

```
var pi = Math.asin(1) * 2;
```

Parameter Name	Description
x	number

10.3 cos

Compute the cosine of an angle

10.3.1 Example

```
var one = Math.cos(0);
```

Parameter Name	Description
angle	angle in radians

10.4 acos

Compute the arc cosine of a number

10.4.1 Example

```
var zero = Math.acos(1);
```

Parameter Name	Description
x	number

10.5 tan

Compute the tangent of an angle

10.5.1 Example

```
var half =  Math.tan(0.463648);
```

Parameter Name	Description
angle	angle in radians

10.6 atan

Compute the arc tangent of a number

10.6.1 Example

```
var pi = Math.atan(1) * 4;
```

Parameter Name	Description
x	number

10.7 sqrt

Compute the square root of a number

10.7.1 Example

```
var three = Math.sqrt(9);
```

Parameter Name	Description
x	number

10.8 log

Compute the natural logarithm of a number

10.8.1 Example

```
var zero = Math.log(1);
```

Parameter Name	Description
x	number

10.9 exp

Compute the base-e exponent of a number

10.9.1 Example

```
var e = Math.exp(1);
```

Parameter Name	Description
x	number

10.10 floor

Compute the largest integral value not greater than the argument

10.10.1 Example

```
var three = Math.floor(3.5);
```

Parameter Name	Description
x	number

10.11 ceil

Compute the smallest integral value not less than the argument

10.11.1 Example

```
var three = Math.ceil(2.5);
```

Parameter Name	Description
x	number

10.12 round

Round to nearest integer, away from zero

10.12.1 Example

```
var three = Math.round(2.7);
var two = Math.round(2.2);
```

Parameter Name	Description
x	number

10.13 abs

Compute the absolute value of an integer

10.13.1 Example

```
var two = Math.abs(-2);
```

Parameter Name	Description
x	number

10.14 atan2

Compute the arc tangent of two variables

10.14.1 Example

```
var pi = Math.atan2(1, 1) * 4;
```

Parameter Name	Description
x	number
y	number

10.15 pow

Power function

10.15.1 Example

```
var hundred = Math.pow(10, 2);
```

Parameter Name	Description
x	number
y	number

CHAPTER 11

Module

11.1 require

Searches for a module and evaluates its code

11.1.1 Example

```
var as = require('assert');
```

Parameter Name	Description
module_name	String containing module name to load

CHAPTER 12

Timer

12.1 setTimeout

Schedule 'cb' to run in ms milliseconds

12.1.1 Example

```
setTimeout(function() { console.log('Timer Expired'); }, 1000);
```

Parameter Name	Description
cb	Callback to be run
ms	Timeout in milliseconds

12.2 setInterval

Schedule 'cb' to run periodically every ms milliseconds

12.2.1 Example

```
setInterval(function() { console.log('.'); }, 1000);
```

Parameter Name	Description
cb	Callback to be run
ms	Period in milliseconds

12.3 clearTimeout

Cancel timeout timer if timeout hadn't expired

12.3.1 Example

```
var to = setTimeout(function() { console.log('timeout'); }, 1000);
clearTimeout(to);
```

Parameter Name	Description
tid	Timer ID to cancel

12.4 clearInterval

Cancel periodic timer

12.4.1 Example

```
var to = setInterval(function() { console.log('.'); }, 1000);
clearInterval(to);
```

Parameter Name	Description
tid	Timer ID to cancel

12.5 getTime

Get number of seconds since system startup

12.5.1 Example

```
console.log('Up for ' + getTime() + ' seconds');
```

CHAPTER 13

Netif

13.1 linkStatus

Get link status

13.1.1 Example

```
var e = new ENC28J60(SPI1, GPIO_PE3, GPIO_PF4);
console.log('link status: ' + e.linkStatus ? 'connected' : 'disconnected')
```

13.2 MACAddrGet

Get Interface MAC Address

13.2.1 Example

```
var e = new ENC28J60(SPI1, GPIO_PE3, GPIO_PF4);
console.log(e.MACAddrGet());
```

13.3 IPAddrGet

Get Interface IP Address

13.3.1 Example

```
var e = new NetifINET('eth0');
console.log(e.IPAddrGet());
```

13.4 onPortChange

Calls 'cb' when link state has changed

13.4.1 Example

```
var e = new ENC28J60(SPI1, GPIO_PE3, GPIO_PF4);
e.onPortChange(function() { console.log('port state changed!'); });
```

Parameter Name	Description
cb	callback function called when link status has changed

13.5 IPConnect

Obtain IP Address

13.5.1 Example

```
var e = new ENC28J60(SPI1, GPIO_PE3, GPIO_PF4);
e.IPConnect();
```

Parameter Name	Description
cb (optional)	callback function called on IP address availablity

13.6 IPDisconnect

Release IP Address

13.6.1 Example

```
var e = new ENC28J60(SPI1, GPIO_PE3, GPIO_PF4);
e.IPDisconnect();
```

13.7 TCPConnect

Connect to a TCP IP:PORT

13.7.1 Example

```
var e = new NetifINET();
e.TCPIPConnect('192.168.1.10', 80, function() { console.log('connected'); });
```

Parameter Name	Description
ip	IP address to connect to
port	tcp port to connect to
cb	callback function called on TCP connectivity

13.8 TCPDisconnect

Release TCP connection

13.8.1 Example

```
var e = new NetifINET();
e.TCPIPConnect('192.168.1.10', 80, function() { console.log('connected'); e.TCPDisconnect() });
```

13.9 onTCPData

Calls 'cb' when TCP data is available

13.9.1 Example

```
var e = new NetifINET();
e.onTCPData(function() { console.log('TCP data ready!'); });
```

Para mete r Name	Desc ript ion
cb	call back func tion call ed when TCP data is avai labl e. Empt y call back remo ves the list ener

13.10 onTCPDisconnect

Calls 'cb' when TCP stream is disconnected

13.10.1 Example

```
var e = new NetifINET();
e.onTCPDisconnect(function() { console.log('TCP disconnected!'); });
```

Para mete r Name	Desc ript ion
cb	call back func tion call ed when TCP stre am is disc onne cted . Empt y call back remo ves the list ener

13.11 TCPWrite

Writes data to the TCP socket

13.11.1 Example

```
var e = new NetifINET();
e.TCPIPConnect('192.168.1.10', 80, function() { e.TCPWrite('GET / HTTP 1.0

'); });
```

Parameter Name	Description
data	Data (byte/array/string/typed array) to be sent

13.12 TCPRead

Reads data from the TCP socket

13.12.1 Example

```
var e = new NetifINET();
e.onTCPData(function() { console.log(n.TCPRead()); });
e.TCPIPConnect('192.168.1.10', 80, function() { e.TCPWrite('GET / HTTP 1.0

'); });
```

13.13 ENC28J60 (constructor)

ENC28J60 Ethernet Constructor

13.13.1 Example

```
var enc = new ENC28J60(SPI0, GPIO_PF0, GPIO_PF1);
console.log('link status: ' + enc.linkStatus() ? 'connected' : 'disconnected');
```

Parameter Name	Description
SPI port	SPI Port
CS	SPI Chip Select Pin
Interrupt	Interrupt Pin

13.14 ESP8266 (constructor)

ESP8266 Wi-Fi Constructor

13.14.1 Example

```
var esp = new ESP8266(function() {
esp.IPConnect(function() { console.log('connected'); });
}, UART4);
```

Parameter Name	Description
cb	Callback to be called when device is ready
Serial port (optional)	Serial Port

13.15 StellarisEth (constructor)

Stellaris Ethernet Object Constructor

13.15.1 Example

```
var enc = new StellarisEth();
console.log('link status: ' + enc.linkStatus() ? 'connected' : 'disconnected');
```

13.16 NetifINET (constructor)

INET Network Interface Object Constructor

13.16.1 Example

```
var  n = new NetifINET('eth0');
console.log('link status: ' + n.linkStatus() ? 'connected' : 'disconnected');
```

Parameter Name	Description
Net Device	Network Interface Name

13.17 LinuxPacketEth (constructor)

Linux Packet Ethernet Object Constructor

13.17.1 Example

```
var  n = new LinuxPacketEth('eth0');
console.log('link status: ' + n.linkStatus() ? 'connected' : 'disconnected');
```

Parameter Name	Description
Net Device	Linux Network Interface Name

13.18 ESP8266_WIFI (constructor)

ESP8266 Wi-Fi Constructor

13.18.1 Example

```
var esp = new ESP8266_WIFI();
esp.IPConnect();
```

CHAPTER 14

console

14.1 set

Sets TinkerPal console to a given serial port

14.1.1 Example

```
console.set(new Serial(UART1));
```

Parameter Name	Description
serial	Serial object

14.2 log

Logs object to the console

14.2.1 Example

```
console.log('Hello World');
```

Parameter Name	Description
object	Object to display

CHAPTER 15

Graphics

15.1 Graphics (constructor)

Graphics Constructor

15.1.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.stringDraw(0, 0, 'Hello World!', 0xffff);
```

Parameter Name	Description
Canvas	Canvas Object

15.2 stringDraw

Prints string on LCD

15.2.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.stringDraw(0, 0, 'Hello World!', 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
str	String to be drawn
color	Color

15.3 circleDraw

Draws a circle on LCD

15.3.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.circleDraw(20, 20, 10, 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
radius	Circle Radius
color	Color

15.4 circleFill

Draws a circle on LCD

15.4.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.circleFill(20, 20, 10, 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
radius	Circle Radius
color	Color

15.5 lineDraw

Draws a line on LCD

15.5.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.lineDraw(10, 10, 20, 20, 0xffff);
```

Parameter Name	Description
x0	X0 coordinate
y0	Y0 coordinate
x1	X1 coordinate
y1	Y1 coordinate
color	Color

15.6 rectDraw

Draws a rectangle on LCD

15.6.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.rectDraw(10, 10, 20, 20, 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
w	Width
h	Height
color	Color

15.7 roundRectDraw

Draws a round rectangle on LCD

15.7.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.roundRectDraw(10, 10, 20, 20, 4, 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
w	Width
h	Height
r	Corner Radius
color	Color

15.8 roundRectFill

Draws a filled round rectangle on LCD

15.8.1 Example

```
var lcd = new Dogs102x6();
var g = new Graphics(lcd);
g.roundRectFill(10, 10, 20, 20, 4, 0xffff);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
w	Width
h	Height
r	Corner Radius
color	Color

15.9 rectFill

Draws a filled rectangle on LCD

15.9.1 Example

```
var lcd = new ST7735();
var g = new Graphics(lcd);
g.rectFill(10, 10, 20, 20, g.RED);
```

Parameter Name	Description
x	X coordinate
y	Y coordinate
w	Width
h	Height
color	Color

CHAPTER 16

Canvas

16.1 pixelDraw

Draw a pixel on the screen

16.1.1 Example

```
var l = new ILI93XX();
l.pixelDraw(10, 10, 1);
```

Parameter Name	Description
x	X Coordinate
y	Y Coordinate
value	Color Value

16.2 fill

Fills the canvas with a color

16.2.1 Example

```
var l = new ILI93XX();
l.fill(0xfe);
```

Parameter Name	Description
value	Color Value

16.3 flip

Publishes stored buffer onto canvas

16.3.1 Example

```
var l = new SSD1306();
l.pixelDraw(1, 1, 1);
l.flip();
```

16.4 SSD1306 (constructor)

SSD1306 Constructor

16.4.1 Example

```
var lcd = new SSD1306();
lcd.pixelDraw(10, 10, 1);
```

16.5 ST7920 (constructor)

ST7920 Constructor

16.5.1 Example

```
var lcd = new ST7920();
lcd.pixelDraw(10, 10, 1);
```

16.6 ST7735 (constructor)

ST7735 Constructor

16.6.1 Example

```
var lcd = new ST7735();
lcd.pixelDraw(10, 10, 1);
```

16.7 DummyCanvas (constructor)

Dummy Canvas Constructor

16.7.1 Example

```
var lcd = new DummyCanvas();
lcd.pixelDraw(10, 10, 1);
```

16.8 SSD1329 (constructor)

SSD1329 Constructor

16.8.1 Example

```
var lcd = new SSD1329();
lcd.pixelDraw(10, 10, 1);
```

16.9 SDLScreen (constructor)

SDL screen Constructor

16.9.1 Example

```
var lcd = new SDLScreen();
lcd.pixelDraw(10, 10, 1);
```

16.10 ILI93XX (constructor)

ili93xx Constructor

16.10.1 Example

```
var lcd = new ILI93XX();
lcd.pixelDraw(10, 10, 1);
```

16.11 Dogs102x6 (constructor)

Dogs102x6 Constructor

16.11.1 Example

```
var lcd = new Dogs102x6();
lcd.pixelDraw(10, 10, 1);
```

16.12 PCD8544 (constructor)

PCD8544 Constructor

16.12.1 Example

```
var lcd = new PCD8544();
lcd.pixelDraw(10, 10, 1);
```

CHAPTER 17

MMC

17.1 MMC (constructor)

MMC Constructor

17.1.1 Example

```
new MMC();
```

CHAPTER 18

SPI

18.1 SPI (constructor)

SPI Object Constructor

18.1.1 Example

```
var s = new SPI(SPI1);
s.send(0xff);
```

Parameter Name	Description
Port ID	HW SPI Port ID

18.2 send

Sends data via SPI bus

18.2.1 Example

```
var s = new SPI(SPI1);
s.send(0x1f);
```

Parameter Name	Description
data	Data to be sent on SPI (integer/array)
cs	Chip select pin (optional)

18.3 receive

Reads data via SPI bus (dummy data is sent)

18.3.1 Example

```
var s = new SPI(SPI1);
var data = s.receive();
```

CHAPTER 19

GPIO

19.1 digitalWrite

Set the digital value of a GPIO or a number of GPIOs

19.1.1 Example

```
digitalWrite(GPIO_PF2, true); /* Turn on PF2 */
digitalWrite([GPIO_PF1, GPIO_PF2, GPIO_PF3], 0x5); /* Turn on PF1 & PF3 */
```

Para mete r Name	Desc ript ion
pin[s]	Sing le GPIO ID or arra y of GPIO IDs
valu e	Bool ean for a sing le GPIO In case of an arra y, valu e is cons ider ed an inte ger wher e thef irst arra y elem ent maps to the MSB

19.2 digitalPulse

Create a digital pulse on a GPIO pin for a given period

19.2.1 Example

```
digitalPulse(GPIO_PF2, true, 0.1);
```

Parameter Name	Description
pin	GPIO pin ID
value	Boolean
ms	Pulse period in milliseconds

19.3 digitalRead

Read the digital state of a GPIO pin or a number of pins

19.3.1 Example

```
var state = digitalRead(GPIO_PF2);
var a = digitalRead([GPIO_PF0, GPIO_PF1, GPIO_PF2
```

Parameter Name	Description
pin[s]	GPIO pin ID or array of GPIO pin IDs

19.4 analogWrite

Set the analog value of a GPIO pin

19.4.1 Example

```
analogWrite(GPIO_PF2, 0.5);
```

Parameter Name	Description
pin	GPIO pin ID
value	Floating point number with range [0-1]
options (Optional)	Object: { freq : }

19.5 analogRead

Read the analog value of a GPIO pin

19.5.1 Example

```
var f = analogRead(GPIO_PF2);
```

Parameter Name	Description
pin	GPIO pin ID

19.6 setWatch

Calls a function whenever the GPIO pin changes state

19.6.1 Example

```
setWatch(function() { console.log('button changed state'); }, GPIO_PF0);
```

Para mete r Name	Desc ript ion
cb	call back func tion :The func tion may rece ive an obje ct of type :{ time stam p: in seco nds, stat e: curr ent pin stat e}
pin	GPIO pin ID
opti ons	opti onal opti ons obje ct of type { qlen : int // numb er of pend ing samp les for proc essi ng smal ler than 128, must be powe r of 2. Defa ult =1 }

CHAPTER 20

Serial

20.1 Serial (constructor)

Serial Constructor

20.1.1 Example

```
var s = new Serial(UART1);
s.print('Hello World!);
```

Para mete r Name	Desc ript ion
UART	HW UART ID
opti ons	opti onal opti ons obje ct of type { baud _ra te : int // Seri al Baud Rate }

20.2 enable

Enable a serial port

20.2.1 Example

```
var s = new Serial(UART1);
s.enable();
```

20.3 disable

Disable a serial port

20.3.1 Example

```
var s = new Serial(UART1);
s.disable();
```

20.4 print

Prints string

20.4.1 Example

```
var s = new Serial(UART1);
s.print('Hello World!');
```

Parameter Name	Description
string	String to be printed to serial port

20.5 write

Writes data to the serial port

20.5.1 Example

```
var s = new Serial(UART1);
s.write(255);
s.write([1,2,3])
```

Parameter Name	Description
data	Data (byte/array/string/typed array) to be sent

20.6 onData

Calls 'cb' when data is available on serial port. If cb is undefined, removes the previously set cb

20.6.1 Example

```
var s = new Serial(UART1);
s.onData(function(e) { s.print(e.data); });
```

Parameter Name	Description
cb	callback function accepting a data object

CHAPTER 21

Boards

The following details the default settings for various devices connected to boards

21.1 STM32F3Discovery

Chipset stm32f303xx

21.1.1 LEDS

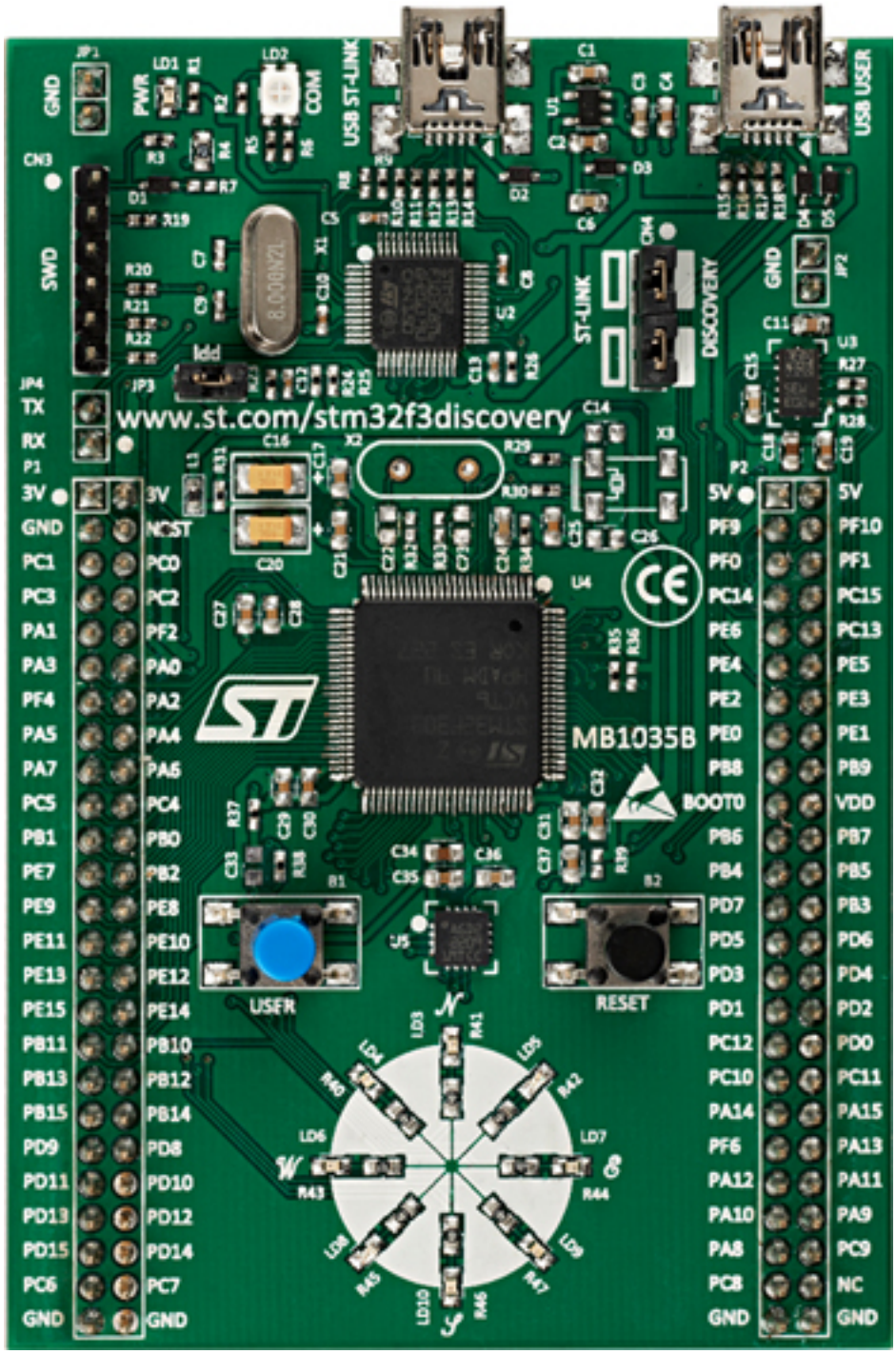
GPIO
PE8
PE9
PE10
PE11
PE12
PE13
PE14
PE15
PF1
PF2
PF3

21.1.2 Console

Console
USART_PORT2

21.2 EK TM4C1294XL (Tiva C Connected Launchpad)

Chipset tm4c1294


www.st.com/stm32f3discovery
MB1035B
USB ST-LINK
USB USER
ST-LINK
DISCOVERY
USER
RESET
SWD
GND
PWR
COM
TX
RX
3V
5V
NCST
PF9
PF0
PC14
PE6
PE4
PE2
PE0
PB8
BOOT0
PB6
PB4
PD7
PD5
PD3
PD1
PC12
PC10
PA14
PF6
PA12
PA10
PA8
PC8
PF10
PF1
PC15
PC13
PE5
PE3
PE1
PB9
VDD
PB7
PB5
PB3
PD6
PD4
PD2
PD0
PC11
PA15
PA13
PA11
PA9
PC9
NC

21.2.1 LEDS

GPIO
PF0
PF4
PN0
PN1

21.2.2 Console

Console
UART0

21.2.3 SSD1306

I2C Port	I2C Address
I2C1	0x78

21.3 Unix based simulator

21.3.1 Console

Console
STDIO_ID

21.3.2 SDL Screen Emulation

Width	Height
320	200

21.4 X86 based simulator

21.4.1 Console

Console
0

21.5 HY 24" STM32F103VET6 based board

Chipset stm32f103vet6

21.5.1 LEDS

GPIO
PC6
PC7
PD13
PD6

21.5.2 Console

Console
USART_PORT2

21.6 ESP8266

21.6.1 Console

Console
UART0

21.7 CC3200-LAUNCHXL (CC3200 Connected Launchpad)

Chipset cc3200

21.7.1 LEDS

GPIO
PB1
PB2
PB3

21.7.2 Console

Console
UART0

21.8 TI MSP430F5529

Chipset msp430f5529

21.8.1 LEDS

GPIO
PA0
PD7

21.8.2 Console

Console
USCIA1

21.8.3 MMC

SPI Port	MOSI Pin	CS Pin
USCIB1	PD1	PC7

21.8.4 ENC28J60

SPI Port	CS Pin	Int. Pin
USCIA0	PC5	PA4

21.8.5 DOGS102X6

SPI Port	RST Pin	CS Pin	CD Pin	Backlight Pin
USCIB1	PE7	PG4	PE6	PG6

21.9 EK LM3S6965

Chipset lm3s6965

21.9.1 Console

Console
UART0

21.9.2 MMC

SPI Port	MOSI Pin	CS Pin
SSI0	PA5	PD0

21.9.3 SSD1329

SPI Port	RST Pin	CS Pin	CD Pin
SSI0	PC6	PA3	PC7


RESET
DEBUG OUT
SELECT
STATUS
PWR
LUMINARYMICRO™
LM3S6965 ETHERNET EVAL BOARD
LUMINARYMICRO™
Luminary Micro
CCP0
Fault
GND
ADC1
ADC3
IDX0
U1RX
U2TX
PHB0
C1+
GND
+3.3V
U0TX
SFSS
SSITX
SDA1
GND
C0-
CCP2
CCP1
ADC0
ADC2
GND
PWM1
U1TX
U2RX
CCP3
PHA0
U0RX
SSICLK
SSIRX
SCL1
GND
+5V
+12V
C1-
PB7
PC3
PHB1
PWM4
SCL0
PWM3
IDX1
LED0
GND
GND
PWM0
GND
C0+
PC2
PHA1
GND
PWM5
SDA0
GND
PWM2
LED1
OSC32
OSC32
+3.3V
USB TO JTAG/ SWD
SD-CARD

21.10 Trinket Pro

21.10.1 LEDS

GPIO
_PB5
_PB6

21.10.2 Console

Console
USART0

21.10.3 SSD1306

I2C Port	I2C Address
0	0x78

21.11 EK LM4F120XL (Stellaris Launchpad)

Chipset lm4f120xl

21.11.1 LEDS

GPIO
PF1
PF2
PF3

21.11.2 Console

Console
UART0

21.11.3 SSD1306

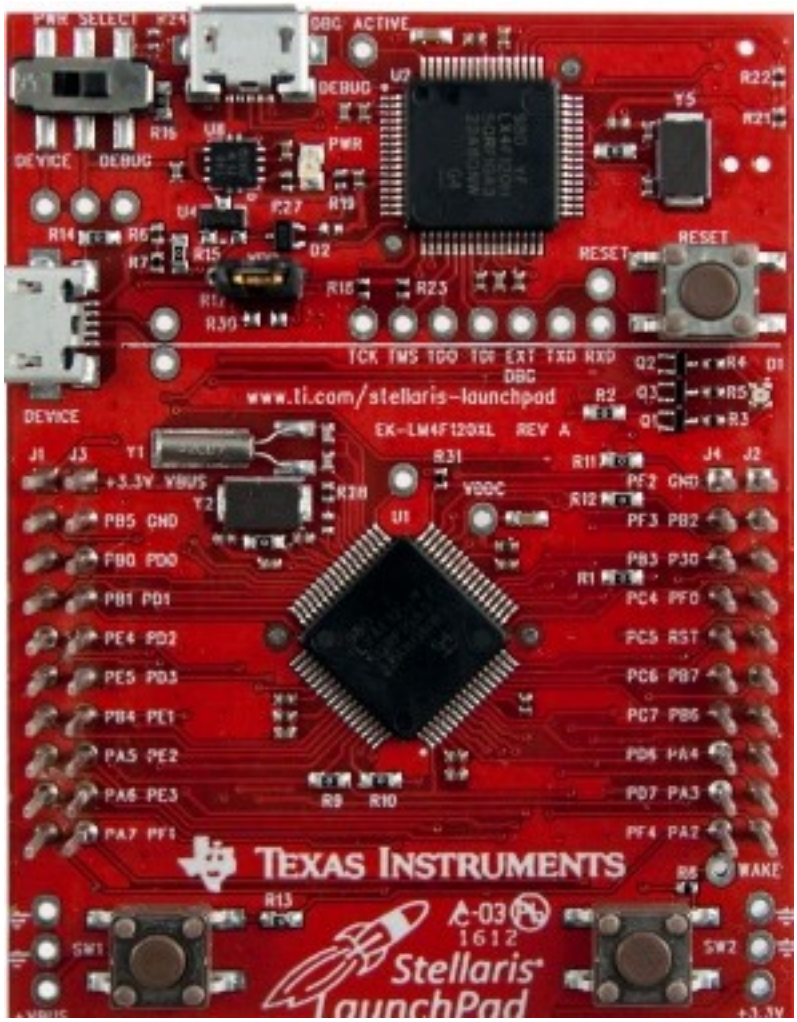
I2C Port	I2C Address
I2C1	0x78

21.11.4 MMC

SPI Port	MOSI Pin	CS Pin
SSI0	PA5	PB6

21.11.5 ENC28J60

SPI Port	CS Pin	Int. Pin
SSI1	PE3	PF4

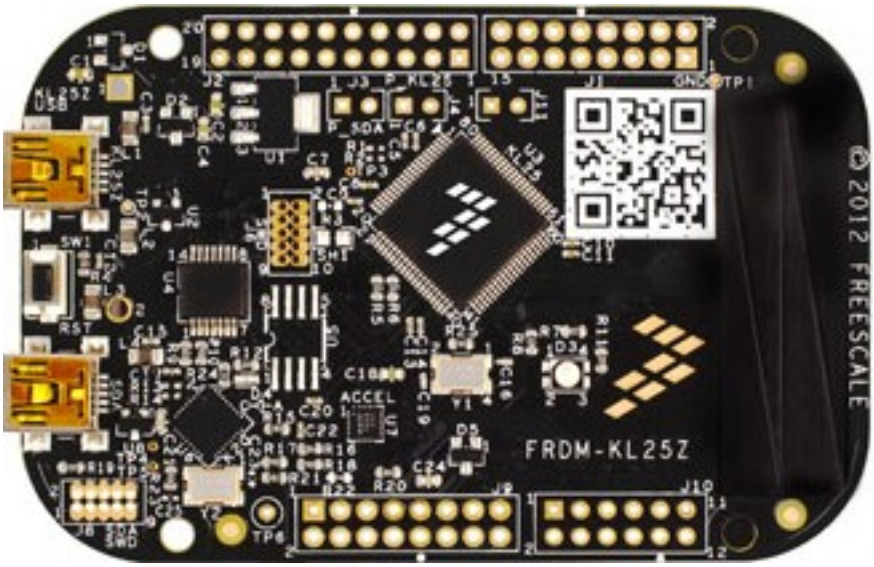


21.12 Freescale FRDM-KL-25Z

Chipset frdm_kl25z

21.12.1 Console

Console
0



21.13 'ARMJishu' 28" STM32F103RB based board

Chipset stm32f103rbt

21.13.1 LEDS

GPIO
PA2
PA3
PB2

21.13.2 Console

Console
USART_PORT1

21.13.3 MMC

SPI Port	MOSI Pin	CS Pin
SPI_PORT1	PA7	PB7

21.14 STM32F4Discovery

Chipset stm32f407xx

21.14.1 Console

Console
USART_PORT2

21.14.2 SSD1306

I2C Port	I2C Address
I2C_PORT1	0x78

21.14.3 ENC28J60

SPI Port	CS Pin	Int. Pin
SPI_PORT1	PC5	PB1

21.15 EK TM4C123GXL (Tiva C Launchpad)

Chipset tm4c123g

21.15.1 LEDS

GPIO
PF1
PF2
PF3

21.15.2 Console

Console
UART0

21.15.3 SSD1306

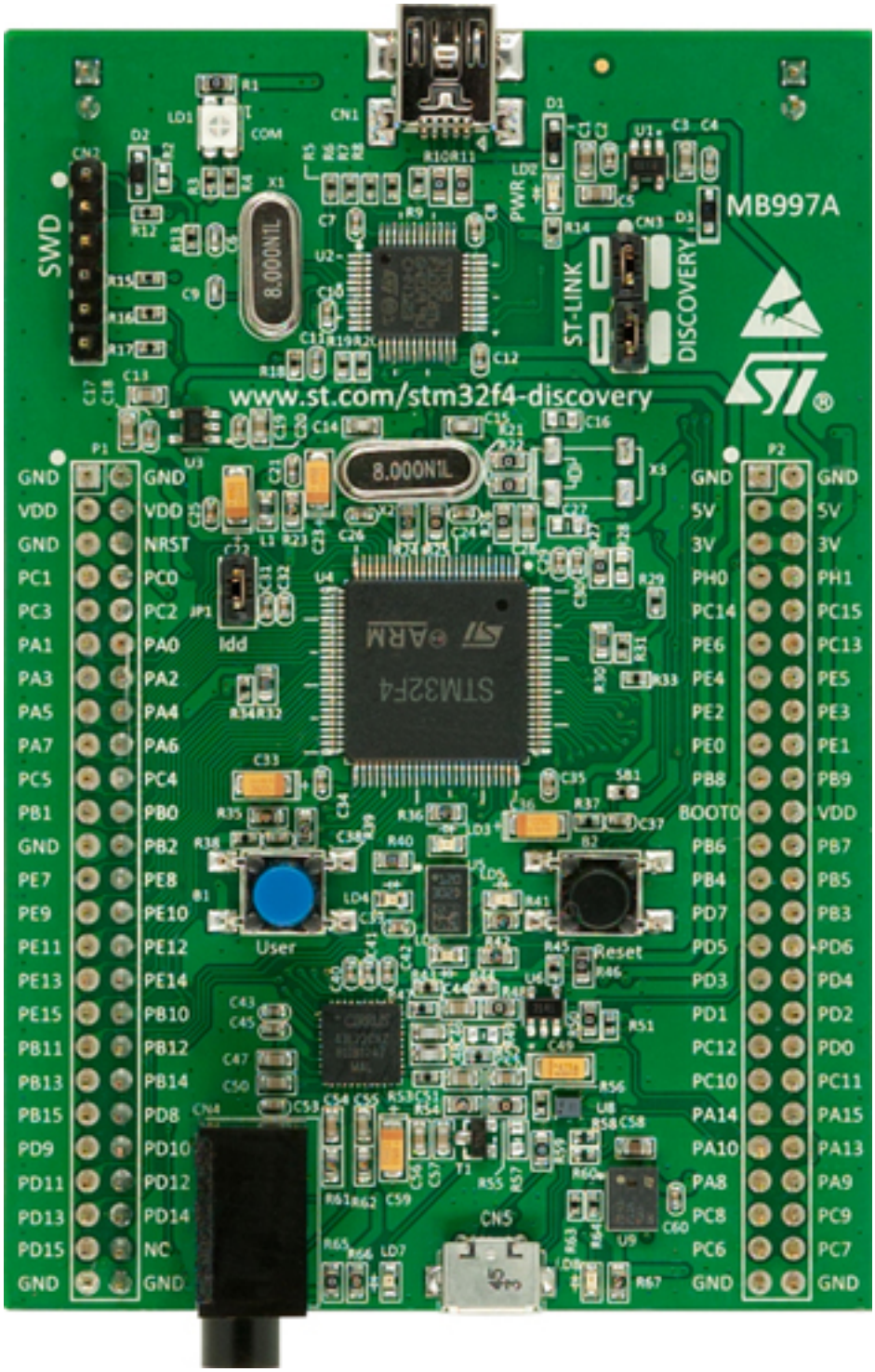
I2C Port	I2C Address
I2C1	0x78

21.15.4 MMC

SPI Port	MOSI Pin	CS Pin
SSI0	PA5	PB6

21.15.5 ENC28J60

SPI Port	CS Pin	Int. Pin
SSI1	PE3	PF4


SWD
MB997A
ST-LINK
DISCOVERY
www.st.com/stm32f4-discovery
8.000N1L
STM32F4
User
Reset
Idd
GND
VDD
NRST
PC0
PC2
PA0
PA2
PA4
PA6
PC4
PB0
PB2
PE8
PE10
PE12
PE14
PB10
PB12
PB14
PD8
PD10
PD12
PD14
NC
PC1
PC3
PA1
PA3
PA5
PA7
PC5
PB1
PE7
PE9
PE11
PE13
PE15
PB11
PB13
PB15
PD9
PD11
PD13
PD15
5V
3V
PH0
PC14
PE6
PE4
PE2
PE0
PB8
BOOT0
PB6
PB4
PD7
PD5
PD3
PD1
PC12
PC10
PA14
PA10
PA8
PC8
PC6
PH1
PC15
PC13
PE5
PE3
PE1
PB9
PB7
PB5
PB3
PD6
PD4
PD2
PD0
PC11
PA15
PA13
PA9
PC9
PC7

21.15.6 ESP8266 Serial Bridge

Serial Port
UART4

21.15.7 PCD8544

SPI Port	RST Pin	CS Pin	CD Pin	Backlight Pin
SSI0	PF3	PB6	PB4	PF2

21.15.8 ST7735

SPI Port	RST Pin	CS Pin	CD Pin	Backlight Pin
SSI0	PF3	PB6	PA6	PF2

21.16 RDK-IDM (LM3S6918)

Chipset lm3s6918

21.16.1 Console

Console
UART1

21.16.2 MMC

SPI Port	MOSI Pin	CS Pin
SSI1	PE3	PE1

21.17 STM32F429IDiscovery

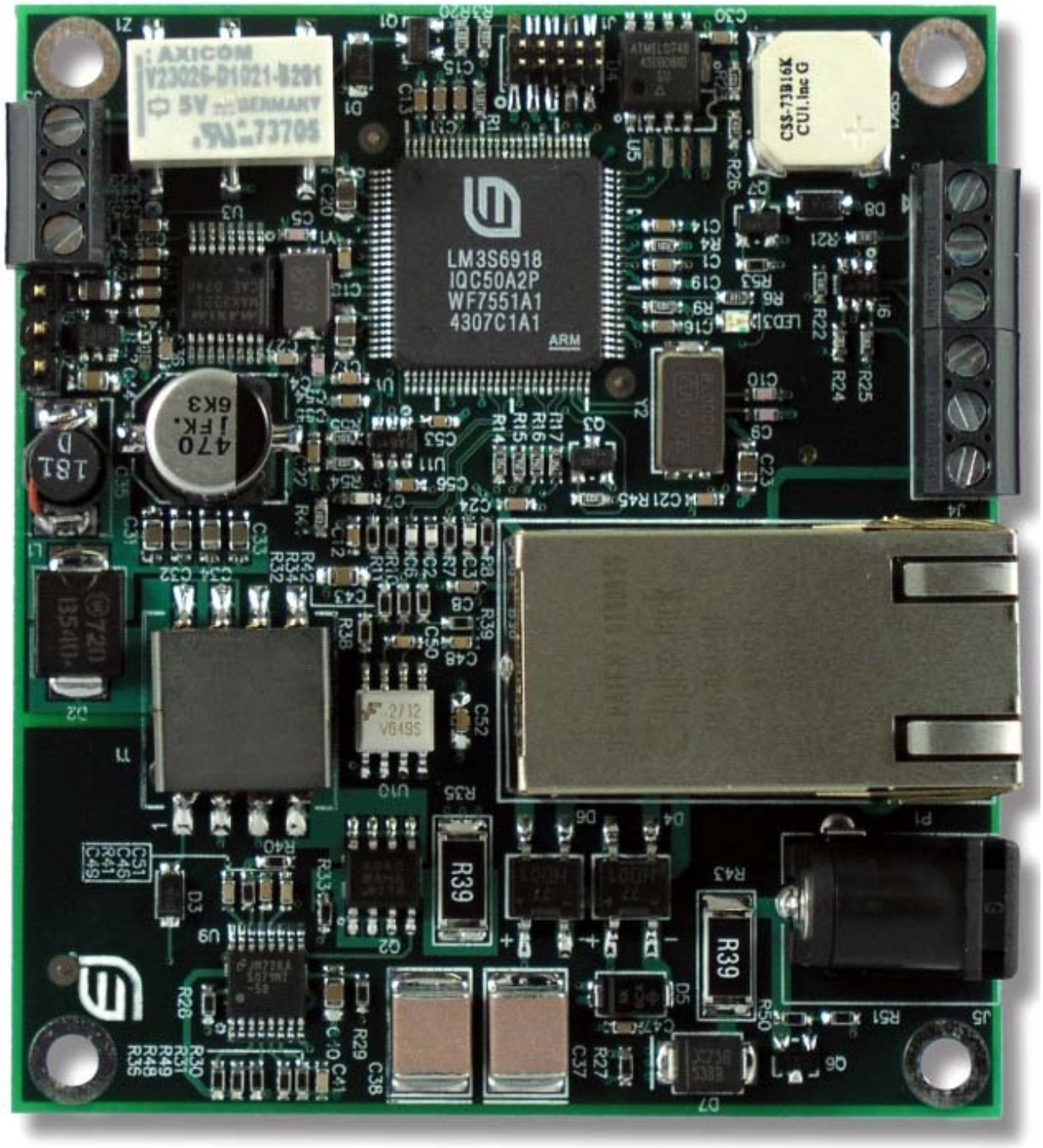
Chipset stm32f429xx

21.17.1 LEDS

GPIO
PG13
PG14

21.17.2 Console

Console
USART_PORT3


AXICOM
LM3S6918
IQC50A2P
WF7551A1
4307C1A1
ARM
R39
R39

CHAPTER 22

Chipset Guide

Chipsets preperties detailed below

22.1 cc3200

22.1.1 Memory Areas

Name	Permissions	Address	Size
SRAM	rwx	0x20004000	192K

22.2 lm4f120xl

22.2.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	256K
SRAM	rwx	0x20000000	32K

22.2.2 UART

UART	RX Pin	TX Pin
UART0	PA0	PA1
UART1	PB0	PB1
UART2	PD6	PD7
UART3	PC6	PC7
UART4	PC4	PC5
UART5	PE4	PE5
UART6	PD4	PD5
UART7	PE0	PE1

22.2.3 I2C

I2C	SCL Pin	SDA Pin
I2C0	PB2	PB3
I2C1	PA6	PA7
I2C2	PE4	PE5
I2C3	PD0	PD1

22.2.4 SSI

SSI	FSS Pin	CLK Pin	RX Pin	TX Pin
SSI0	PA3	PA2	PA4	PA5
SSI1	PD1	PD0	PD2	PD3

22.2.5 USBD

DP Pin	DM Pin
PD5	PD4

22.3 tm4c123g

22.3.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	256K
SRAM	rwx	0x20000000	32K

22.3.2 UART

UART	RX Pin	TX Pin
UART0	PA0	PA1
UART1	PB0	PB1
UART2	PD6	PD7
UART3	PC6	PC7
UART4	PC4	PC5
UART5	PE4	PE5
UART6	PD4	PD5
UART7	PE0	PE1

22.3.3 I2C

I2C	SCL Pin	SDA Pin
I2C0	PB2	PB3
I2C1	PA6	PA7
I2C2	PE4	PE5
I2C3	PD0	PD1

22.3.4 SSI

SSI	FSS Pin	CLK Pin	RX Pin	TX Pin
SSI0	PA3	PA2	PA4	PA5
SSI1	PD1	PD0	PD2	PD3

22.3.5 USBD

DP Pin	DM Pin
PD5	PD4

22.4 tm4c1294

22.4.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	1024K
SRAM	rwx	0x20000000	256K

22.4.2 UART

UART	RX Pin	TX Pin
UART0	PA0	PA1
UART1	PB0	PB1
UART2	PA6	PA7
UART3	PA4	PA5
UART4	PA2	PA3
UART5	PC6	PC7
UART6	PP0	PP1
UART7	PC4	PC5

22.4.3 I2C

I2C	SCL Pin	SDA Pin
I2C0	PB2	PB3
I2C1	PG0	PG1
I2C2	PN5	PN4
I2C3	PK4	PK5
I2C4	PK6	PK7
I2C5	PB0	PB1
I2C6	PA6	PA7
I2C7	PA4	PA5
I2C8	PA2	PA3
I2C9	PA0	PA1

22.4.4 SSI

SSI	FSS Pin	CLK Pin	RX Pin	TX Pin
SSI0	PA3	PA2	PA4	PA5
SSI1	PB4	PB5	PE4	PE5
SSI2	PD2	PD3	PD1	PD0

22.4.5 USBD

DP Pin	DM Pin
PL6	PL7

22.5 lm3s6918

22.5.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	256K
SRAM	rwx	0x20000000	64K

22.5.2 UART

UART	RX Pin	TX Pin
UART0	PA0	PA1
UART1	PD2	PD3

22.5.3 SSI

SSI	FSS Pin	CLK Pin	RX Pin	TX Pin
SSI0	PA3	PA2	PA4	PA5
SSI1	PE1	PE0	PE2	PE3

22.6 lm3s6965

22.6.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	256K
SRAM	rwx	0x20000000	64K

22.6.2 UART

UART	RX Pin	TX Pin
UART0	PA0	PA1

22.6.3 SSI

SSI	FSS Pin	CLK Pin	RX Pin	TX Pin
SSI0	PA2	PA2	PA4	PA5

22.7 frdm_kl25z

22.7.1 Memory Areas

Name	Permissions	Address	Size
INTERRUPTS	rx	0x00000000	0xC0
FLASHCFG	rx	0x00000400	0x10
FLASH	rx	0x00000800	128K - 0x800
SRAM	rwx	0x1FFFF000	16K

22.8 stm32f103rct6

22.8.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x08000000	256K
SRAM	rwx	0x20000000	48K

22.8.2 UART

UART	RX Pin	TX Pin
USART1	PA10	PA9
USART2	PA3	PA2
USART3	PB11	PB10

22.8.3 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB13	PB14	PB15

22.9 stm32f103vet6

22.9.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x08000000	512K
SRAM	rwx	0x20000000	64K

22.9.2 UART

UART	RX Pin	TX Pin
USART1	PA10	PA9
USART2	PA3	PA2
USART3	PB11	PB10

22.9.3 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB13	PB14	PB15

22.10 stm32f103rbt

22.10.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x08000000	128K
SRAM	rwx	0x20000000	20K

22.10.2 UART

UART	RX Pin	TX Pin
USART1	PA10	PA9
USART2	PA3	PA2
USART3	PB11	PB10

22.10.3 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB13	PB14	PB15

22.11 stm32f303xx

22.11.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x00000000	256K
SRAM	rwx	0x20000000	40K

22.11.2 UART

UART	RX Pin	TX Pin
USART1	PB7	PB6
USART2	PA3	PA2
USART3	PB11	PB10
UART4	PB11	PB10
UART5	PD2	PC12

22.11.3 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB13	PB14	PB15
SPI3	PC10	PC11	PC12

22.12 stm32f407xx

22.12.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x08000000	1024K
SRAM	rwx	0x20000000	128K
CCM	rwx	0x10000000	64K

22.12.2 UART

UART	RX Pin	TX Pin
USART1	PA10	PA9
USART2	PA3	PA2
USART3	PB11	PB10
UART4	PA1	PA0
UART5	PD2	PC12
USART6	PC7	PC6

22.12.3 I2C

I2C	SCL Pin	SDA Pin
I2C1	PB8	PB9
I2C2	PB10	PB11
I2C3	PA8	PC9

22.12.4 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB10	PC2	PC3
SPI3	PC10	PC11	PC12

22.13 stm32f429xx

22.13.1 Memory Areas

Name	Permissions	Address	Size
FLASH	rx	0x08000000	2048K
SRAM	rwx	0x20000000	192K
CCM	rwx	0x10000000	64K

22.13.2 UART

UART	RX Pin	TX Pin
USART1	PA10	PA9
USART2	PA3	PA2
USART3	PB11	PB10
UART4	PA1	PA0
UART5	PD2	PC12
USART6	PC7	PC6
UART7	PE7	PE9
UART8	PE0	PE1

22.13.3 SPI

SPI	CLK Pin	MISO Pin	MOSI Pin
SPI1	PA5	PA6	PA7
SPI2	PB10	PC2	PC3
SPI3	PC10	PC11	PC12
SPI4	PE2	PE5	PE6
SPI5	PF7	PF8	PF9
SPI6	PG13	PG12	PG14

22.14 msp430f5529

22.14.1 USCI

USCI	RX Pin	TX Pin	CLK Pin
USCIA0	PC4	PC3	PB7
USCIA1	PD5	PD4	PD0
USCIB0	PC1	PC0	PC2
USCIB1	PD2	PD1	PD3