Connecting iOS devices to RS-232 serial devices

Talk at Cocoaheads

Aachen 08/30/2012

Serial devices

Serial lines

Connectivity

iOS-Devices

Redpark cable and SDK

Serial devices: Yes or No?

Serial cable

VGA cable

Serial: History

Historical serial lines are derived from phone lines. There were 2 standards, the 25-pins Centronics and the 9-Pins DE-9, later the Mini-DIN 8 on Apple, NeXT and SGI.

Pro:

- Old standard, was widespread
- Speed is slow to moderate
- reliable
- Easy to build (or solder)

Cons

- Old standard
- Wired

Serial: Usage

... Apple dropped serial lines with Mac OSX.

- Dial-up modems
- GPS receivers (typically NMEA 0183 at 4,800 bit/s)
- Bar code scanners
- OLder phones and cameras
- Uninterruptible power supply
- ... and more

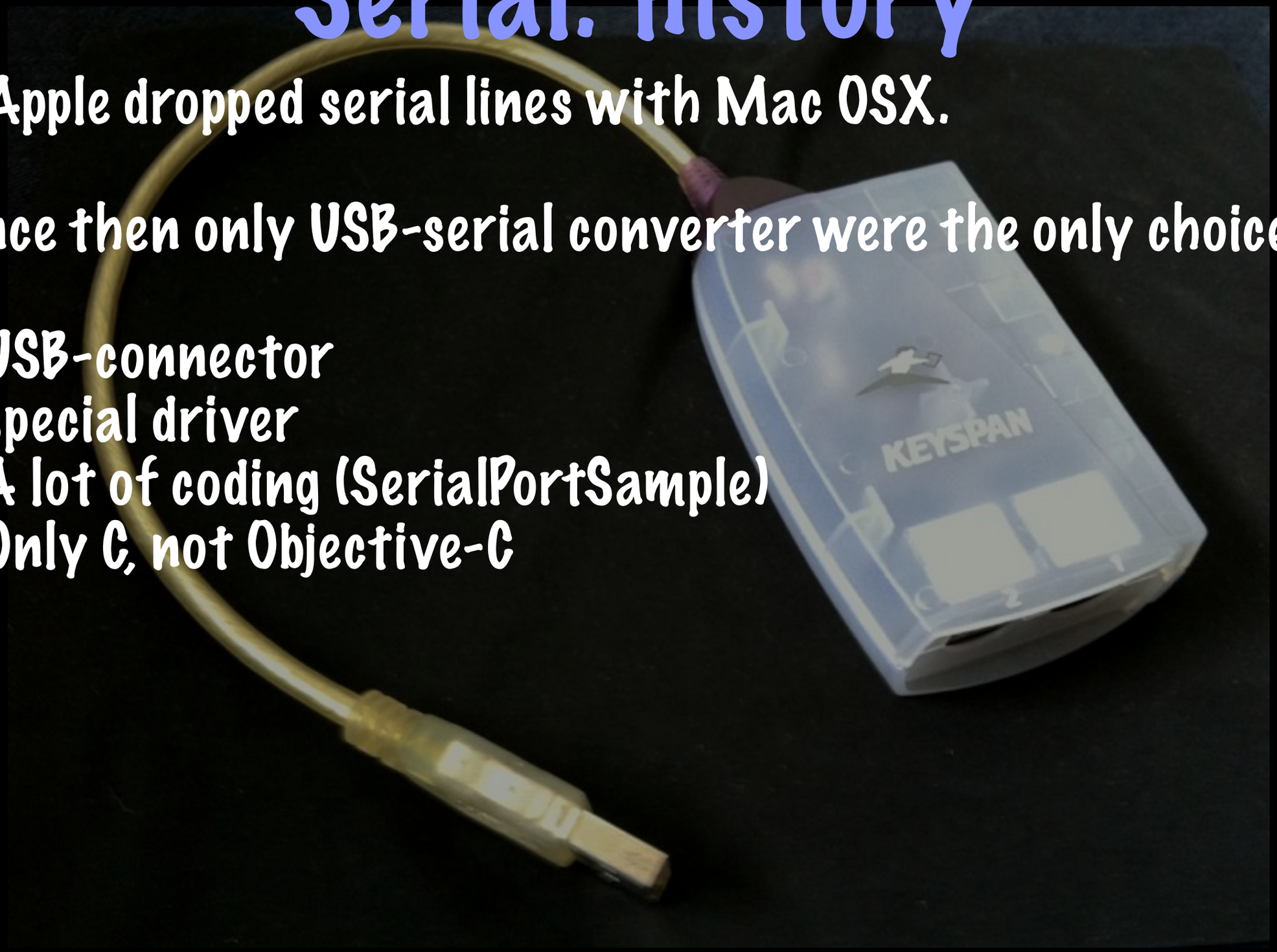
Generally devices are all types of sensors, and controls.
Sounds like... robotics, maybe.

Serial: History

... Apple dropped serial lines with Mac OSX.

Since then only USB-serial converter were the only choice.

- USB-connector
- special driver
- A lot of coding (SerialPortSample)
- Only C, not Objective-C



Serial: History

... Apple dropped serial lines with Mac OSX.

Biggest problem: What chip is in that e.g. board from sparkfun?

The drivers were a nightmare. Chips from different manufactureres are used, FTDI, Prolific and others.
Their drivers had to be installed. Some of them were buggy, e.g. did not handle multiple instances or unfortunate naming conventions. Updates are behind several releases by Apple.

iOS: Connectivity

iOS-Devices are connected!

Wireless:
- Wifi
- Bluetooth
- Cellular transmission (Edge, 1...NG, LTE)

Wired:
- Dock connector cable

The single dock connector is the only access point for wired connections.

iOS: Serial

Apple has no serial line on any iOS device.

- Sensors
- External GPS receivers with greater accuracy
- External visual Scanners
- External Controls
- ... and maybe more

Redpark Cable

Redpark SDK

The Redpark Serial cables are used only in conjunction with the SDK from Redpark.

From their website:
"Together these tools enable hobbyist, education and enterprise developers to write iOS apps that communicate with serial devices. These apps may be deployed for private use at home, at school or in an office.
Under current Apple policy this cable may not be used with apps sold on the App Store."

Refrain: No App Store, if used.

Redpark SDK

Simple project: Just build something around the templates provided by Redpark.

Usage:

- Include the framework
- Instantiate the manager
- Instantiate a delegate object, probably a kind of controller
- Implement the delegate-methods

Done. Because it is a boxed system, there is no confusion wit different drivers or more than one instances.

Showcase

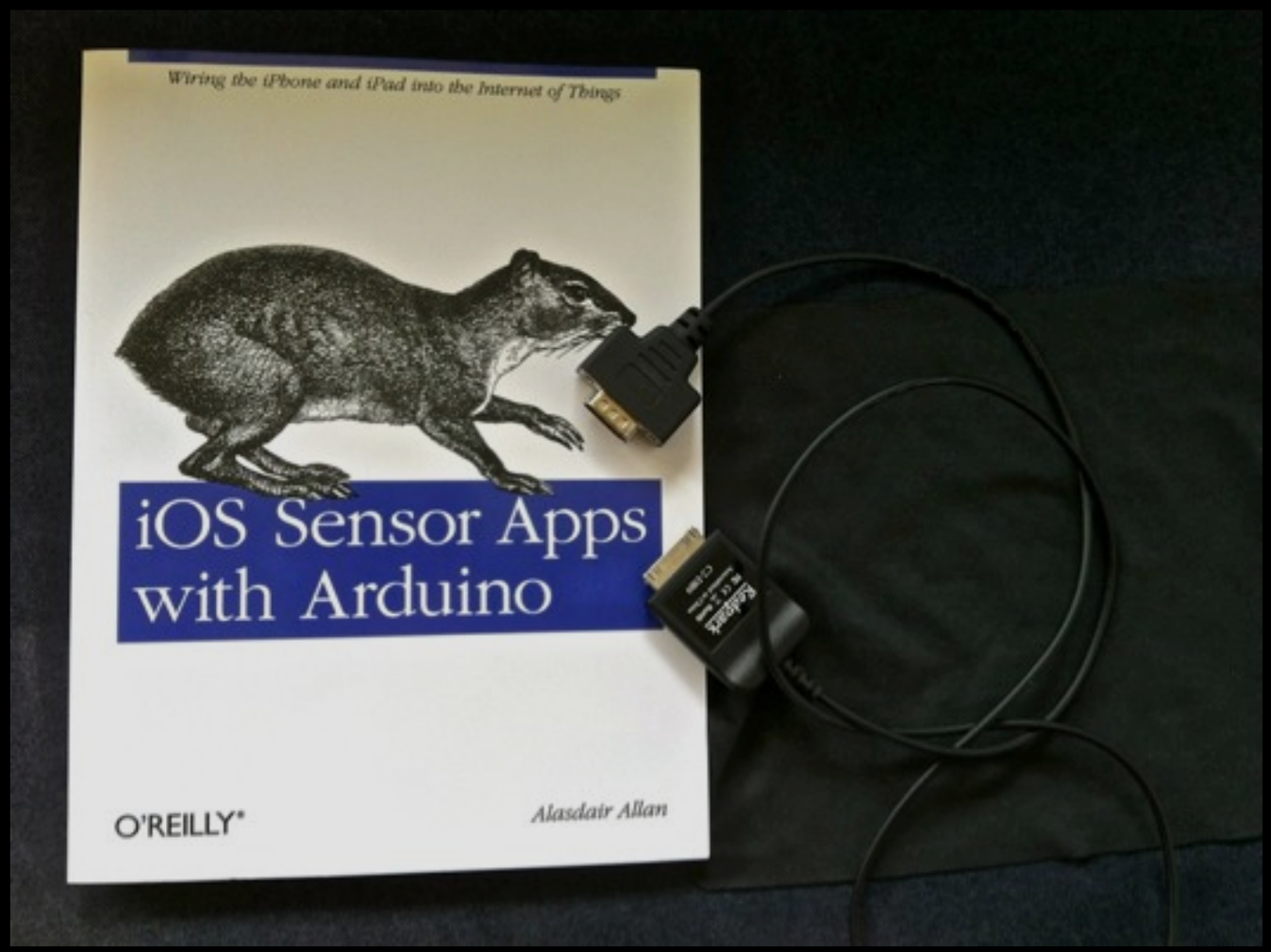
```
////////////////////////////////////////////////////////////////////////
    // Create and initialize our RedparkSerialCable Manager for communicating
    // with the serial cable.
    rscMgr = [[RscMgr alloc] init];

    // Set this as our delegate so we can receive evevnts
   [rscMgr setDelegate:self];

    // For convenience the strings which populate the various port config screens
    // are built up from a plist.
   portConfigTableData = [self readPlist:TABLE_DATA_PLIST];
   portConfigKeys = [[portConfigTableData allKeys]
                          sortedArrayUsingSelector:@selector
(localizedCaseInsensitiveCompare:)];
   [portConfigKeys retain];
```

Redpark SDK: delegate

```
@protocol RscMgrDelegate  <NSObject>

// Redpark Serial Cable has been connected and/or application moved to foreground.
// protocol is the string which matched from the protocol list passed to initWithProtocol:
- (void) cableConnected:(NSString *)protocol;

// Redpark Serial Cable was disconnected and/or application moved to background
- (void) cableDisconnected;

// serial port status has changed
// user can call getModemStatus or getPortStatus to get current state
- (void) portStatusChanged;

// bytes are available to be read (user calls read:)
- (void) readBytesAvailable:(UInt32)length;

@optional
// called when a response is received to a getPortConfig call
- (void) didReceivePortConfig;

// GPS Cable only - called with result when loop test completes.
- (void) didGpsLoopTest:(BOOL)pass;

@end
```

Serial devices

There is a special theme:



Thank you.