

Sensor Network Communication (Lecture Notes)

Spencer Russell

Hardware Platforms

Raw Microcontrollers

- Main considerations
 - Integrated RF or separate radio chip?
 - 8Bit, 16Bit, 32Bit
 - Clock Speed
 - Flash (code space)
 - RAM
 - peripheral config
 - toolchain
 - supply V (1.8V, 3.3V, 5V)
 - power consumption (should sleep in <1uA)
- PIC
 - don't recommend
 - confusing memory model
 - crappy tools
 - inexpensive
 - used to be very popular, mostly comes up in older designs
- Atmel AVR
 - lots of support in hobbiest community
 - gcc support
- ST Micro (STM8, STM32)
 - STM32 is ARM core (supports GCC)
 - toolchains suck for stm8
- TI MSP430
 - 16Bit
 - gcc support
 - also good hobbiest community

Arduino

- big gorilla
- based on AVR Micro
- ubiquitous
- “shield” system for add-ons
- lots of libraries
- program in C++
- super easy to flash (bootloader)

Embedded Linux Single-Board-Computers

- Intel Galileo
 - i586 processor
 - ethernet
- Raspberry Pi
 - ARMv6 processor
 - ethernet
- BeagleBone Black
 - ARMv7 processor
 - ethernet
- Intel Edison
 - more like a system-on-module
 - Atom processor
 - super compact
 - onboard wifi and bluetooth
 - no ethernet
 - low-power sleep mode

Digital Sensors

analog-to-digital conversion

- Reference voltage
- sampling (nyquist)
- sample at twice your highest frequency and low-pass first
- num of bits
- V_{ref} is $2^N - 1$

Representing Digital information

- ASCII
- Binary

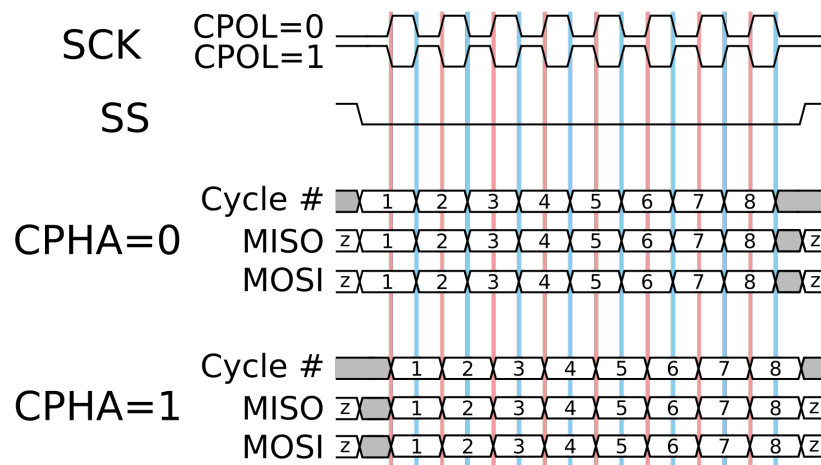
Transmitting digital information

- Serial vs. Parallel
- 1-way vs. half-duplex vs. full-duplex
- baud rate vs. Bps vs bps
- data order and endianness
- asynch vs synch
- differential signaling
- buffer over/underruns
- multi-drop vs. point-to-point

RS232 / UART

- often just called “serial”
- single-ended, common ground -> short runs
- +/- 3-15V for ones and zeros
- full duplex (if both TX and RX are used and software supports)
- 1-to-1 (usually, but could fan out TX to multiple RX)

SPI (Serial Peripheral Interface)



- MISO, MOSI, SCK, SCL
- short distance
- high speed
- full duplex
- Watch out for Mode (polarity and phase)
- simple protocol, easy to see on logic analyzer

I2C

- short-distance
- needs pull-up resistors
- master-slave, half-duplex

RS485

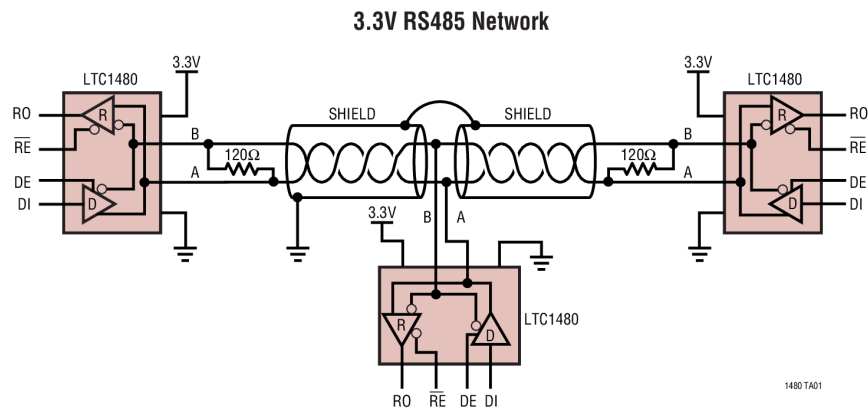
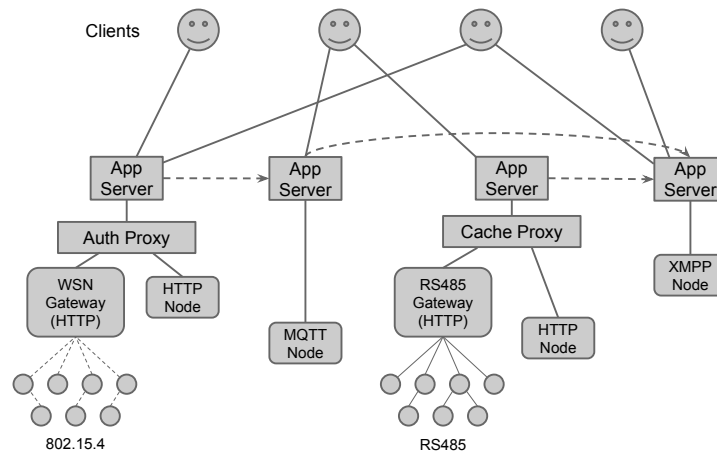


Figure 1: <http://www.linear.com/product/LTC1480>

- differential (good for long distances ~ 1mile)
- half-duplex
- master-slave

Radio Communication

- Point-to-Point RF Links
 - generally operate as UART replacement



- Mesh Networking
 - Atmel
 - ZigBee
 - 6LoWPAN
 - * Basically the use of IPv6 on top of 802.15.4
 - * requires some [translation](#) to accomodate spec incompatibilities between the layers
- Bluetooth
 - relatively short-range, high bandwidth applications such as audio streaming
 - not good for very low-power devices because of slow wake/connect time
- [Bluetooth LE](#)
 - much better wake/connect times, making it suitable for low-power long-running battery-operated devices
 - shorter range than classic BT
 - Star-topology (no meshing)
- WiFi
 - Like bluetooth, unsuitable for very low-power devices because of very long wake/connect time
 - I predict an increase in wifi-connected IoT devices for line-powered devices though, as chip costs go down and they integrate more of the TCP/IP Stack

The Internet

Basic IP Protocols

- XMPP
 - XML-based protocol originally used for Jabber
 - Has a rich extension system which has been used to define an application layer protocol for sensor data and device control
 - [sensor data](#)
 - [provisioning](#)
 - [control](#)
 - [concentrator](#)
- MQTT
 - [Backed](#) by Cisco/IBM/others
 - Being worked on as an OASIS standard
 - Intended to be usable on low-power/low-bandwidth devices ([8bit, 256KB RAM](#)), but still may be out-of-reach for extremely power/cost sensitive devices
 - used by facebook for messaging
- CoAP
 - maps HTTP to binary protocol
 - quite a bit of support in the research community
 - nice simple overview at [17](#)
- HTTP / REST
- WebSockets
 - easy javascript integration
 - widely used and understood
- AMQP
 - broker-based message queue
- ZeroMQ
 - lightweight
 - up and running quickly
 - best behind a firewall

Hosted Services

- PubNub
 - Pretty thin message queue wrapper

- lots of support for different platforms (node, Arduino, etc.)
- SparkFun [Phant](#)
 - very simple
 - open source
 - all streams public
 - HTTP API
 - supports MQTT [16](#)
- [xively](#) (formally Cosm (formally Patchube))
 - supports access via REST, as well as Sockets, WebSockets, and MQTT
 - Real-time connections use payload format modeled after HTTP
- ChainAPI