Peripherals

These lecture notes created by Mr. James B. (Jim) Carlson, NCSU

Overview

- Liquid Crystal Display (LCD)
- Touchscreen
- Interfacing Methods
- Sensors
- Mass Storage

- *Use google and wikipedia to get more information*

LCD

- Basics
  - Sandwich of polarizer, glass, electrode, liquid crystal, electrode, glass, polarizer (90 deg. from first)
  - Liquid crystal is twisted to change polarization of light passing through, so it makes it through both polarizers (light/clear)
  - Apply voltage to electrodes to reduce twist, reducing polarization and hence darkening the display element (dark/opaque)
- Backlighting
  - LED – low voltage (e.g. 1.5 – 4 V)
  - CCFL (cold-cathode flourescent) – high voltage (e.g. 100V)

Arranging the Dots

- Segmented (e.g. 7-segment, 16-segment)
- Dot matrix
  - Text only
  - Text and graphics

Passive vs. Active

- Becomes an issue when you have a multiplexed display (don't have a wire per pixel or segment)
- Passive – Rely on LC to stay in same state of twistiness before it is refreshed next. Slow response time, limited contrast
- Active – Build transistors onto the display right next to the electrode to retain the state between refreshes

Driving the LCD

- Direct
  - Dedicated wire per segment. Just apply a voltage.
  - Lots of wires and pins
- Multiplexing options
  - Need to turn on and off pixels frequently enough to keep them from fading
  - Multiplexed segments
    - Connections
      - Segment 1-40 (5x8 segments per character location)
      - Common 1-16 (8x2 = 16 character locations
    - Select the character with the Common line, then turn on the appropriate segments
    - Use a controller chip (e.g. HD44780, KS0066)
    - Could also use MCU to do it in software or with DMA

More Multiplexing

- Rectangular matrix of rows and columns
  - Select the row, then turn on the appropriate pixels (columns)
  - Shift registers are used to simplify interface, so timing evolbed from raster scanning of CRTs
    - For each frame, VSync signal resets vertical counter
      - For each row, HSync signal resets horizontal shift register and advances vertical counter
        - » For each pixel, Clock signal shifts some data into horizontal (column driver) shift register. E.g. 1 pixel's worth (18 or 24 bits)
  - Use a controller chip (e.g. T6963, SED1335)

Resistive Touchscreen

- Other technologies available too (capacitive, acoustic, optical, etc.)
- Sandwich
  - Conductive flexible membrane (has an electrode if 5-wire)
  - Spacers
  - Substrate (has 4 electrodes (X+, X-, Y+, Y-) along edges)
  - Display
- Using it
  - Determine X position
    - Apply V+ to X+, V- to X-
    - Measure analog voltage at electrode (5-wire) or at Y+ and Y-
  - Determine Y position
    - Apply V+ to Y+, V- to Y-
    - Measure analog voltage at electrode (5-wire) or at X+ and X-
  - Linearize X and Y based on calibration
- More Information
  - http://www.edn.com/archives/1995/110995/23dfcov.htm

Interfacing Methods

- Analog – use ADC or comparator
- Digital
  - Pulse width modulation
    - Measure width of pulse with timer or interrupts
  - RS232
    - Use UART
    - Bit-bang with GPIO if not available
  - synchronous (clocked) serial
    - Use UART in clocked serial mode
    - Bit-bang with GPIO if not available
  - parallel
    - Use GPIO port

Clocked (Synchronous) Serial Interfacing

- Basic ideas
  - Send data serially between chips to reduce pin count and PCB complexity
  - Provide *a clock* with the serial data to make interfacing easy
- Example protocols
  - SPI – serial peripheral interconnect.
  - I^2C^{TM} – inter-integrated circuit bus.

SPI – Serial Peripheral Interface

- Based on shift registers
- Example: AD7877
- Signals to get bytes in and out
  - DCLK – Data clock
  - DIN – Data into peripheral device.
    - Is read at rising edge of DCLK (Must be ready 12 ns before)
    - Also called MOSI: master out, slave in
  - DOUT – Data out of peripheral device.
    - Is valid 16 ns after falling edge of DCLK
    - Also called MISO: master in, slave out
  - Need to provide individual chip-selects if multiple SPI devices share DCLK/DIN/DOUT

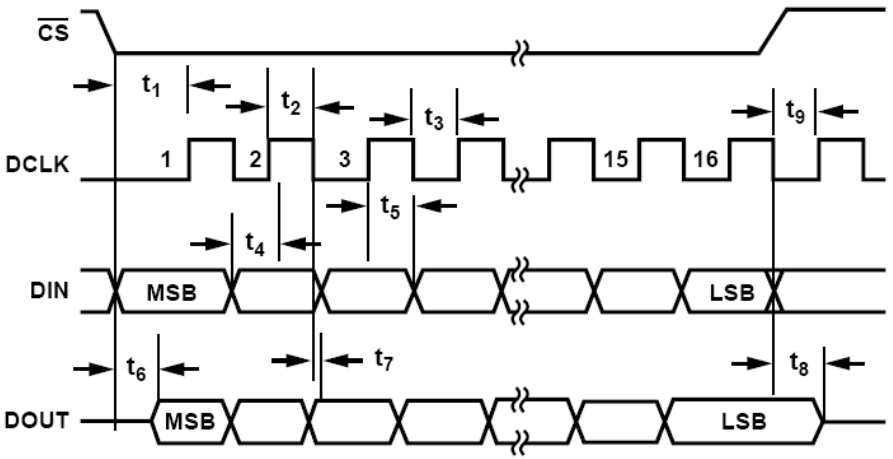


Figure 2. Detailed Timing Diagram

Other Comments

- Other SPI-based devices will have different protocol details
  - Command length
  - Register address length
  - Register addresses

I^2C^{TM} – inter-integrated circuit bus.

- Full-fledged protocol created by Philips (now NXP Semiconductor, see document UM10204 for details)
- TWI – two wire interface – generic name for I^2C^{TM}
- Signals
  - SCK – serial clock
  - SDA – serial data – bidirectional
- 100-400 kbps
- Foundation for other protocols (SMBus and PMBus)

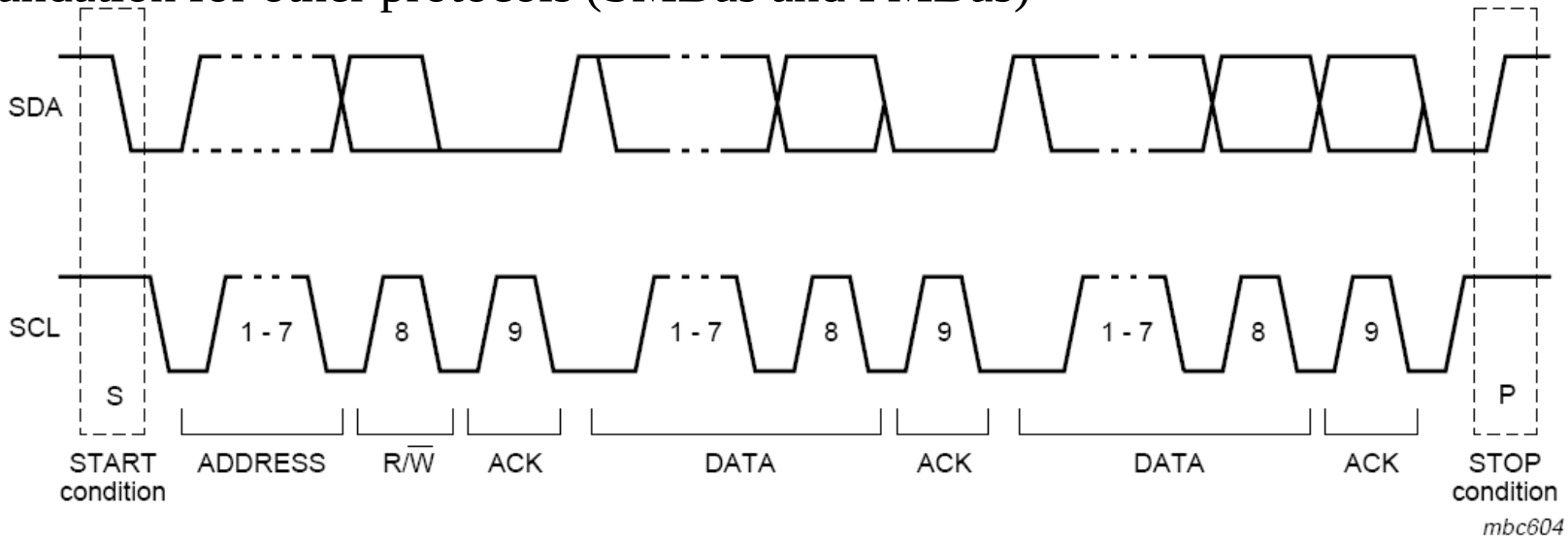


Fig 9. A complete data transfer

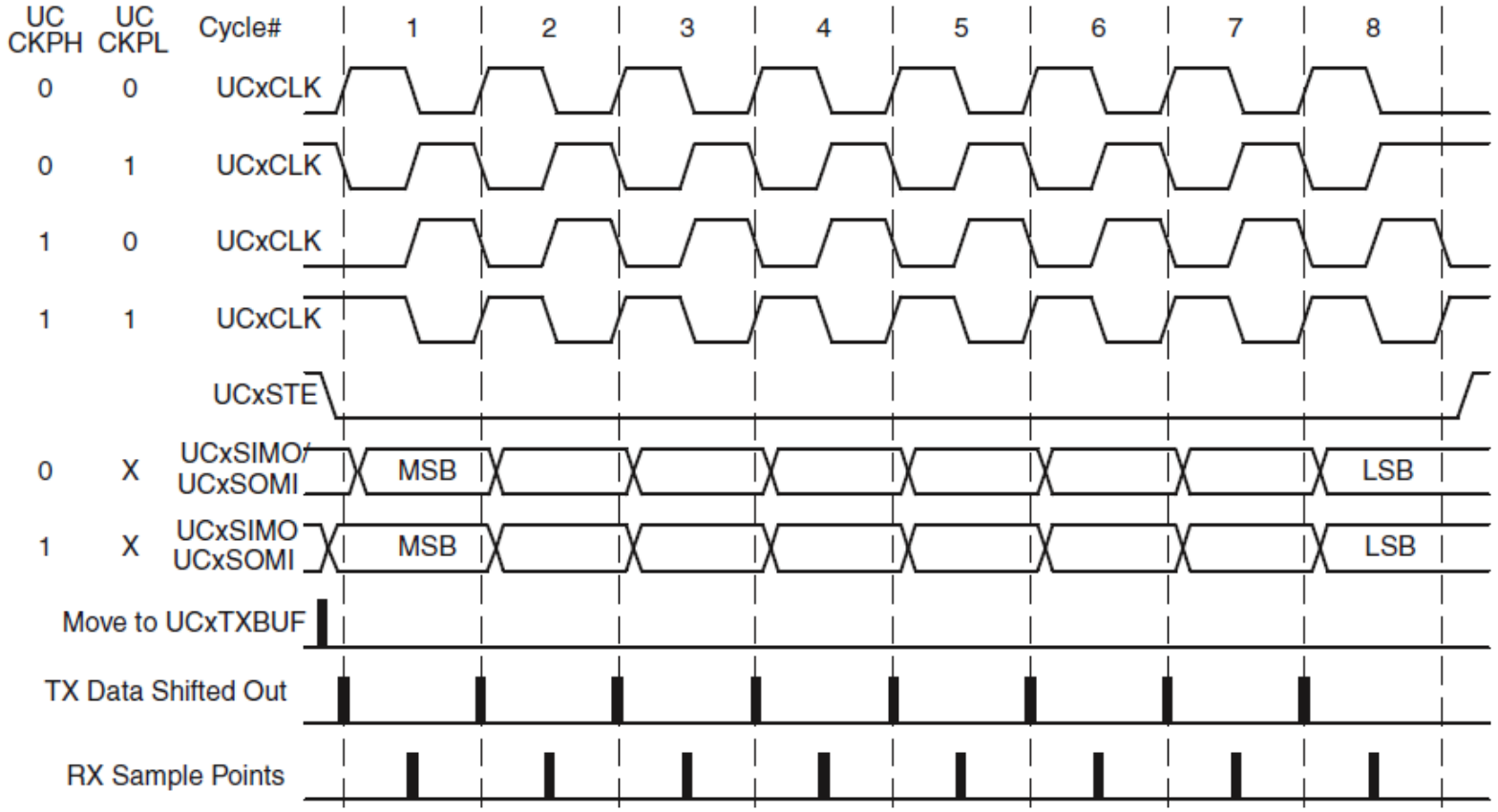
SPI – Serial Peripheral Interface

- In SPI mode, serial data is transmitted and received by multiple devices using a shared clock provided by the master. An additional pin controlled by the master, UCxSTE, is provided to enable a device to receive and transmit data.
- Three or four signals are used for SPI data exchange:
- Master mode: UCxSIMO is the data output line.
- Master mode: UCxSOMI is the data input line.
- Master mode: UCxCLK is an output.
- Master mode: Chip Select is a user defined port pin.

- Master controls clock for both data transfer from Master to Slave and from Slave to master.

SPI – Serial Peripheral Interface

- **Serial Clock Polarity and Phase**
- The polarity and phase of UCxCLK are independently configured via the UCCKPL and UCCKPH control bits of the eUSCI.

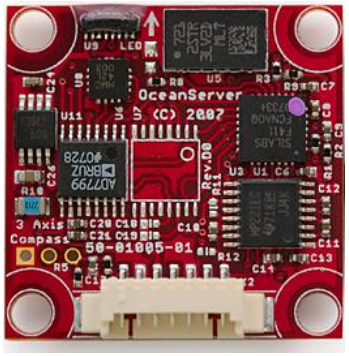


Sensors, and Applications

- Remember these?
- Temperature
  - Thermometer (do you have a fever?)
  - Thermostat for building, fridge, freezer
  - Car engine controller
  - Chemical reaction monitor
  - Safety (e.g. microprocessor processor thermal management)
- Light (or infrared or ultraviolet) intensity
  - Digital camera
  - IR remote control receiver
  - Tanning bed
  - UV monitor
- Rotary position
  - Wind gauge
  - Knobs
  - Compass
- Humidity
- Proximity
- Pressure
  - Blood pressure monitor
  - Altimeter
  - Car engine controller
  - Scuba dive computer
  - Tsunami detector
- Acceleration
  - Air bag controller
  - Vehicle stability
  - Video game remote
- Mechanical strain
- Pressure
- Capacitive (touch)
- Other
  - Touch screen controller
  - EKG, EEG
  - Breathalyzer

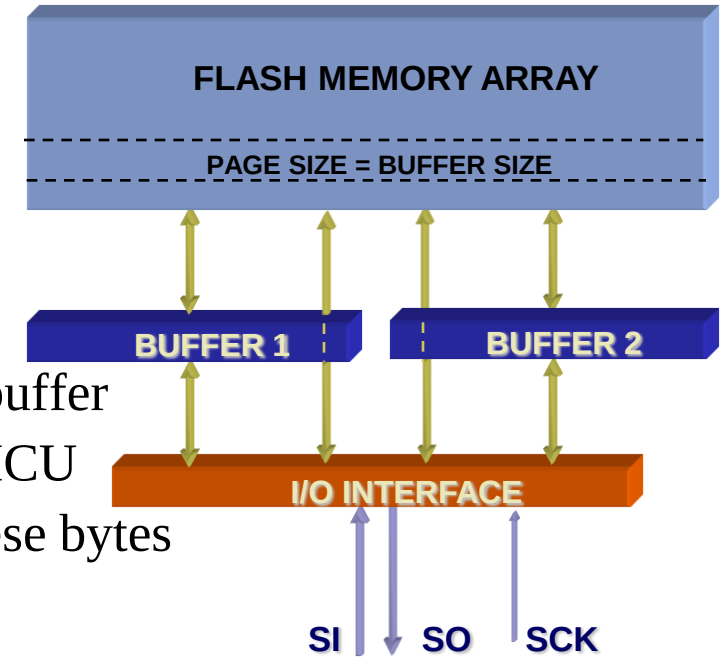
Example Sensors

- Compass
  - RS232 interface
  - 1 degree accuracy
  - Tilt-compensated
  - Model rocket, robot, etc.
- Accelerometer
  - Analog or PWM output
  - 2-10 G ranges available
  - Model rocket, tilt, etc.
- Ultrasonic rangefinder
  - RS232, PWM and analog outputs
  - 0-255 inch range, 1 inch resolution
  - 20Hz update rate
- Inertia Measurement System
  - Acceleration in X,Y,Z directions
  - Rotation around X,Y,Z axes (gyros)
  - Temperature
  - Federal funds discount rate ;-)
  - Logic-level RS232 serial interface
- *Images courtesy SparkFun Electronics*



Fixed Mass Storage of Non-Volatile Data

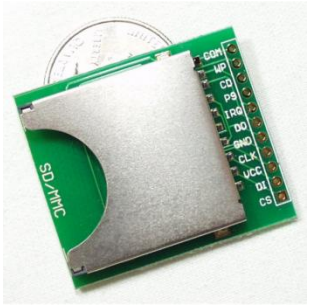
- Use MCU's flash program memory. 384KB!
  - Programming interface
    - Write these bytes starting at this address
    - Read from address in MCU memory
- Add external flash memory chips to circuit board
  - Atmel DataFlash chips
  - SPI-based interface
  - Programming interface
    - Read page N from flash into an SRAM buffer
    - Transfer SRAM buffer contents to the MCU
    - Load the SRAM buffer (or part) with these bytes
    - Write SRAM buffer to Flash page N



A Brief Digression – File Systems

- What's there
  - Lots of bytes, each individually addressable
- What we want
  - Individual files of different lengths
    - Ability to create, delete, append, overwrite these files
  - Perhaps even a hierarchical collection of folders/directories
- File system
  - Provides abstractions to give us the above
  - Index (directory table) in each directory tells about each item (file or directory)
    - Name, starting address, length, access date, etc.
- Examples
  - FAT12, FAT32, NTFS

Removable Mass Storage 1



- Secure Digital card
  - Based on MMC (multimedia card) – Toshiba added encryption hardware to make it Secure
  - SD cards support MMC's SPI interface mode
    - How to use MMC/SDC: http://elm-chan.org/docs/mmc/mmc_e.html
    - Also Maxim application note AN3969.pdf
  - Interface is low-level
    - Read, write block (e.g. 512 bytes)
    - Doesn't directly support a file system
      - User must provide this (e.g. ChaN, FatFs, TinyFatFs)
      - Or else use your own scheme for tracking used/free blocks

Removable Mass Storage 2

- USB Mass storage device (flash drive)
  - Use an interface module to avoid having to hack USB support
  - Vinculum VDIP1 and VDIP2 modules
    - http://www.vinculum.com/prd_vdip1.html
    - USB peripheral device (e.g. mass storage)
    - MCU interface: UART, SPI or FIFO (parallel)
  - Interface includes high-level features:
    - Has built in-support for FAT file systems
      - Open, close, delete files,
      - List files in directory, get sizes
    - Append data, read data, seek offset
    - Lots of other features