

ECE2049: LECTURE 13

OFFICE HOURS

0

TODAY

- REVIEW HW6
- DIGITAL INTERFACES

- TODAY: 5-7 PM EDT (NICK)

- WED - 2-4 PM (JONATHAN)
- 5-7 PM (NICK)
EDT

- THURS - 2-4 PM, 5-7 PM EDT
(NICK)

ADMINISTRIVIA

- HW6: DUE TODAY AT 11:59 PM EDT

LAB 2

- I RECOMMEND YOU HAVE STAGES 1A/1B DONE EARLY THIS WEEK

- SIGNOFF DUE MONDAY (7/12) BY 7PM EDT, REPORT 7/12 BY 11:59 PM EDT

EXAM 2, ~~OUT~~

- OUT. ON THURSDAY (7/8), DUE MON (7/12) BY 11:59 PM EDT

- SEE CANVAS ANNOUNCEMENT FOR DETAILS

NEXT LECTURE: - REVIEW PRACTICE EXAM

- COURSE DISCUSSION (YOUR THOUGHTS/FEEDBACK ABOUT THE COURSE)

- THIS IS OUR FINAL WEEK

IF YOU HAVE OUTSTANDING LABS OTHER THAN LAB 2, AND YOU HAVE NOT TALKED WITH ME RECENTLY, YOU SHOULD CONTACT ME ASAP.

- YOU MUST ATTEMPT ALL LABS TO PASS THE COURSE

- I CAN BE FLEXIBLE, BUT I NEED TO HEAR FROM YOU SOON

1. ~~Temp~~ $V_{TEMP} = \underbrace{.00475}_{\substack{V/^{\circ}C \\ -60^{\circ}C - 90^{\circ}C}} (TEMP) + 1.29V$

a. $-60^{\circ}C$ $V_{TEMP} = .00475(-60^{\circ}C) + 1.29V$
 $= 1.005V$

$90^{\circ}C$ $V_{TEMP} = 1.7175V$

b. REFERENCE VOLTAGE?

1.5V, 2.5V, 3.3V

$V_{REF} = 2.5V$ TO COVER THE WHOLE REGION.
 COVER

c. SMALLEST CHANGE IN TEMP?

$R_{LS} = \frac{2.5V}{2^{12}} = .610 mV/BIT$ $^{\circ}C/BIT$

$\left(\frac{.610 mV}{1 BIT} \right) \left(\frac{1^{\circ}C}{.00475V} \right) = \boxed{.1284^{\circ}C/BIT}$

FROM SLOPE OF SENSOR EQUATION

ANOTHER WAY

EX.

CODE 0 $\Rightarrow 0V$

$0V = .00475(T_0) + 1.29V$

$T_0 = -271.579^{\circ}C$

CODE 1 $\Rightarrow .610 mV$

$.610 mV = .00475(T_1) + 1.29V$

$T_1 = -271.45^{\circ}C$

$T_{DIFF} = T_1 - T_0 = \boxed{.1284^{\circ}C/BIT}$

$$d. T_{EMP} = 24.8^{\circ}C$$

SENSOR \rightarrow VOLTS

$$V_{OUT} = .00475(24.8^{\circ}C) + 1.29V$$

FOR OUR ADC:

$$= 1.4078V$$

$$V_{REF+} = 2.5V$$

$$V_{REF-} = 0V$$

$$CODE = \left[\left(\frac{V_{IN} - V_{REF-}}{V_{REF+} - V_{REF-}} \right) (2^N - 1) \right]$$

$$= \frac{V_{IN}}{2.5V} (2^{12} - 1) = \frac{1.4078V}{2.5V} (2^{12} - 1)$$

$$= [2305.97]$$

$$= \boxed{2305}$$

$$e. CODE = 2415$$

$$2415 = \frac{V_{IN} - V_{REF-}}{V_{REF+} - V_{REF-}} (2^N - 1) \quad 2415 = \frac{V_{IN}}{2.5V} (2^{12} - 1)$$

$$V_{IN} = 1.47V$$

$$\textcircled{1} V_{IN} = (CODE) \left(\frac{2.5V}{2^{12} - 1} \right)$$

$$\textcircled{1} 1.47V = .00475(^{\circ}C) + 1.29V$$

$$^{\circ}C = \boxed{38.73V}$$

$$\textcircled{2} T_{EMP} = \frac{(V_{IN} - 1.29)}{.00475}$$

f. CODE \Rightarrow VOLTS \rightarrow DEGREES.

USE $\textcircled{1}$ AND $\textcircled{2}$

$$2. \tau_{\text{INT}} = 25 \text{ ms} / \text{INTERUPT}$$

$$\text{TARGET} = 500 \text{ MS}$$

$$\frac{500 \text{ MS}}{25 \text{ MS}} = \underline{\underline{20 \text{ INTERUPTS}}}$$

3. a. INTERUPTS!

$$\text{AAA} = 1.5\text{V}$$

b. BATTERY CAPACITY = 1100 mA·h

2 AAA IN SERIES. ~~V_{CC}~~ = 3V

(i) ALWAYS IN ACTIVE MODE?

1. STATE ASSUMPTIONS.

$$I_{AM} = 0.36 \text{ mA}$$

ASSUME TYPICAL VALUES.

$$\text{RUNTIME} = \frac{1100 \text{ mA}\cdot\text{h}}{0.36 \text{ mA}} = \boxed{3055 \text{ HOURS}}$$

(ii) LPM0 FOR 84% OF TIME?
AM FOR THE REST

$$I_{LPM0} = 83 \mu\text{A} \quad \text{ASSUME } 25^\circ\text{C}$$

$$\begin{aligned} I_{AVG} &= (0.84)(83 \mu\text{A}) + (1-0.84)(0.36 \text{ mA}) \\ &= \del{127.3} 127.3 \mu\text{A} \end{aligned}$$

$$\text{RUNTIME} = \frac{1100 \text{ mA}\cdot\text{h}}{127.3 \mu\text{A}} = 8639 \text{ HOURS} \approx \boxed{359 \text{ DAYS}}$$

Module 11. Intro to Digital Interfaces

Digital Interface: Connection between two or more digital devices.

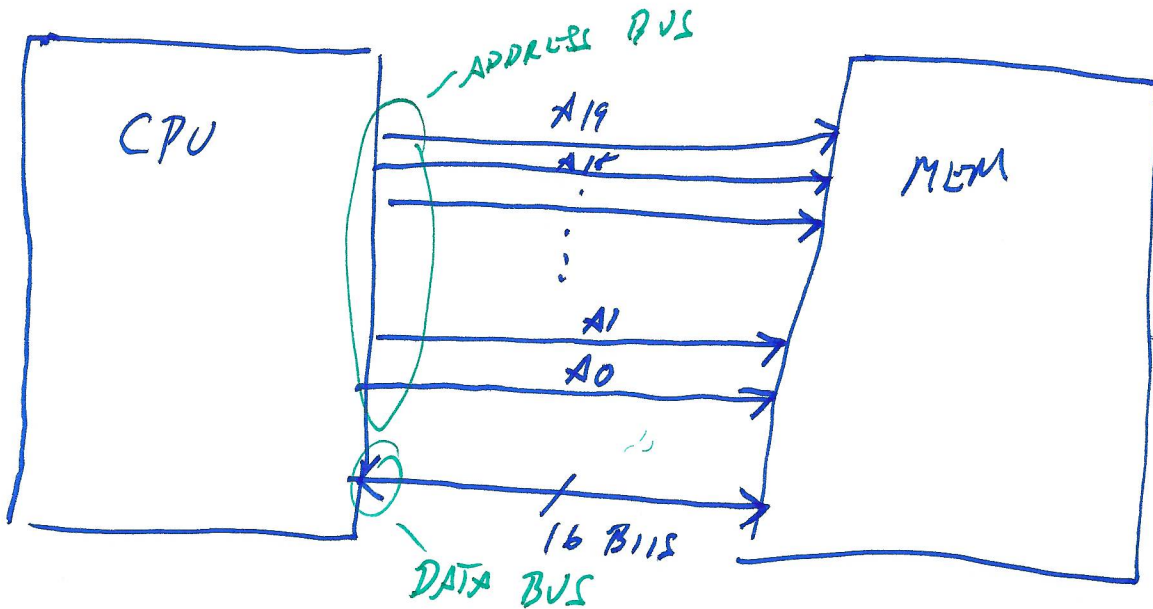
*INSTEAD OF A VOLTAGE SCALE,
ENCODE INFORMATION IN DIGITAL LOGIC
(JUST LIKE A CPU)*

Types of Digital Interfaces

There are two classifications for how data is transmitted in a digital interface: serial and parallel

Parallel Interfaces = Each bit has its own electrical connection (interconnect, trace, wire)

>> Advantages = Fast, easier to synchronize (1 clock edge transfers all bits together)



>> Used almost exclusively inside a CPU (or other) chip

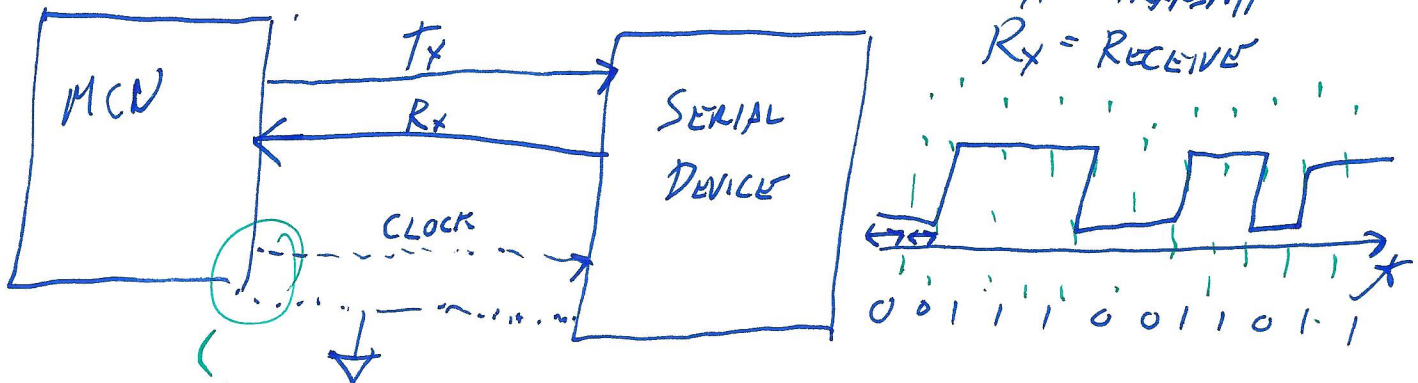
- MSP430 (MEMORY BUS: CPU TO MEMORY/
PERIPHERALS)
- INSIDE THE CPU ITSELF
(REGISTERS TO ALU)

>> Disadvantage = Each bit must have its own electrical connection

- INEFFICIENT IN "REAL ESTATE"

- HARD TO ADD ~~THE~~ SO MANY PINS TO A CHIP, OR EVEN INSIDE A CHIP.

Serial Interfaces – Bits sent one after another along a single connection



ONLY ON SOME

>> Used almost exclusively to make connections off-chip (and off-computer, through the Internet, out to the Mars Rover, etc.)

>> Advantages = Simpler/fewer connections between CPU and peripheral (2-4 lines)

Examples include:

Device Select/Enable (CS)

Synchronizing CLK (SCLK)

Data Line(s) (SDI and SDO)

Common ground (GND): Often already established if devices are on the same board or IC

USB 2.0 480 MBPS

USB 3.0: 5 GBPS

MODERN INTERFACES USE BOTH

STRATEGIES: MULTIPLE SERIAL LINES!

>> "Connection" = PCB trace, wire, RF, acoustic, optical, etc.

>> Disadvantages = "Slower", more complicated synchronization, potential timing issues

(↓ PER CLOCK CYCLE

MODERN SERIAL INTERFACES ARE VERY

FAST MEGABITS/GIGABITS PER SECOND.

How are digital interfaces implemented?

Most microcontrollers contain hardware peripherals that implement the interface in hardware. On the MSP430, we have hardware peripherals to implement a few types of serial interfaces.

On the MSP430: Universal Serial Communications Interface (UCSI)

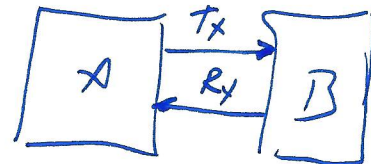
- >> Basically acts as a parallel-to-serial and serial-to-parallel converter
- >> Most modern microprocessors/microcontrollers will have built-in Serial interface
 - MSP430F5229 has a total of four, in two types
- >> Role of Serial interfaces has grown with growing sophistication and speed of serial links (SPI and I²C to USB and others)

We will discuss three types of interfaces: UART, SPI, and I2C. There are many more types of digital interfaces!

What kind of information is exchanged?

Just like in a CPU, information in a digital interface is represented in bits. The interface defines the manner in which bits are transmitted:

- WE CAN SEND WHATEVER WE WANT



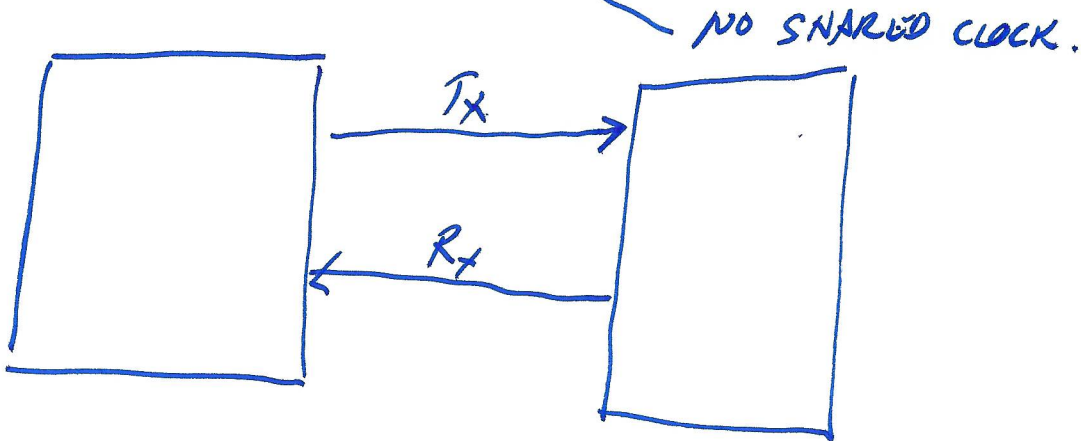
- IN PRACTICE THERE ARE DEFINED FORMATS FOR HOW DATA IS SENT.

↳ (DATASHEETS!)

Types of Interfaces

UART

- >> UART: Universal Asynchronous Receiver/Transmitter
- >> UART mode configures basic 2-wire asynchronous serial communications
- >> Connect to UCSI with 2 external pins (MSP430: UCA_xRXD and UCA_xTXD)



>> Not synchronous (no shared clock) = Asynchronous

>> To use serial communications both devices must know data format and baud rate

- These are set using UARTs control registers
- Implies make data format & baud rate decisions at design time



UART: Fundamental Parameters

In order to use UART for an application, you need to determine the following parameters:

- Start Bit = 1 bit (“low”)
- Data Bits = 7 or 8 bits
- Parity = Even, Odd, or None
 - >> Even Parity = 1 when number of 1's including parity is even
 - >> Odd Parity = 1 when number of 1's including parity is odd
- Stop bit(s) = 1 or 2 bits (“high”)
- Baud rate (bits/sec): Common rates are 200, 2400, 9600, 19200, 115200

CAN BE USED FOR (WEAK) ERROR CHECKING.

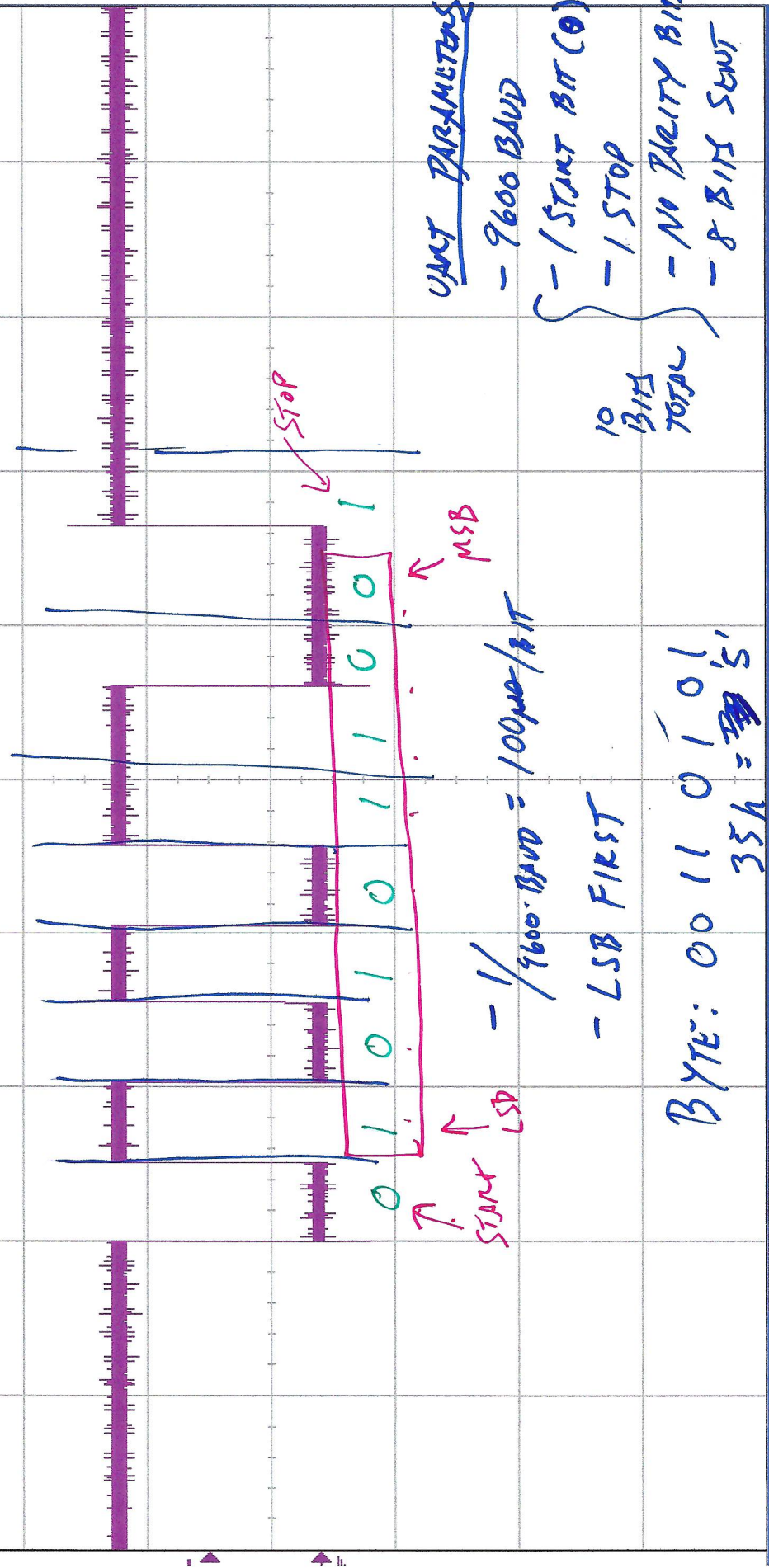
RS-232 – “Old standby” for serial format → Actually is a specific standard with associated voltage ranges and baud rates now often misused to mean any asynchronous serial communication

-->Data sent Least Significant Bit (LSB) first**

** Most UARTs send asynchronous serial data LSB first (because RS-232 is LSB first) but MSP430F5229 UCSI_A is configurable and can be set to send MSB first, so that it can be compatible with various types of serial interfaces.

EXAMPLE UART TRANSMISSION

USB device installed as "/drive0".



New file name = scope_0

Press to go to /drive0

Spell s

Enter

Delete Character

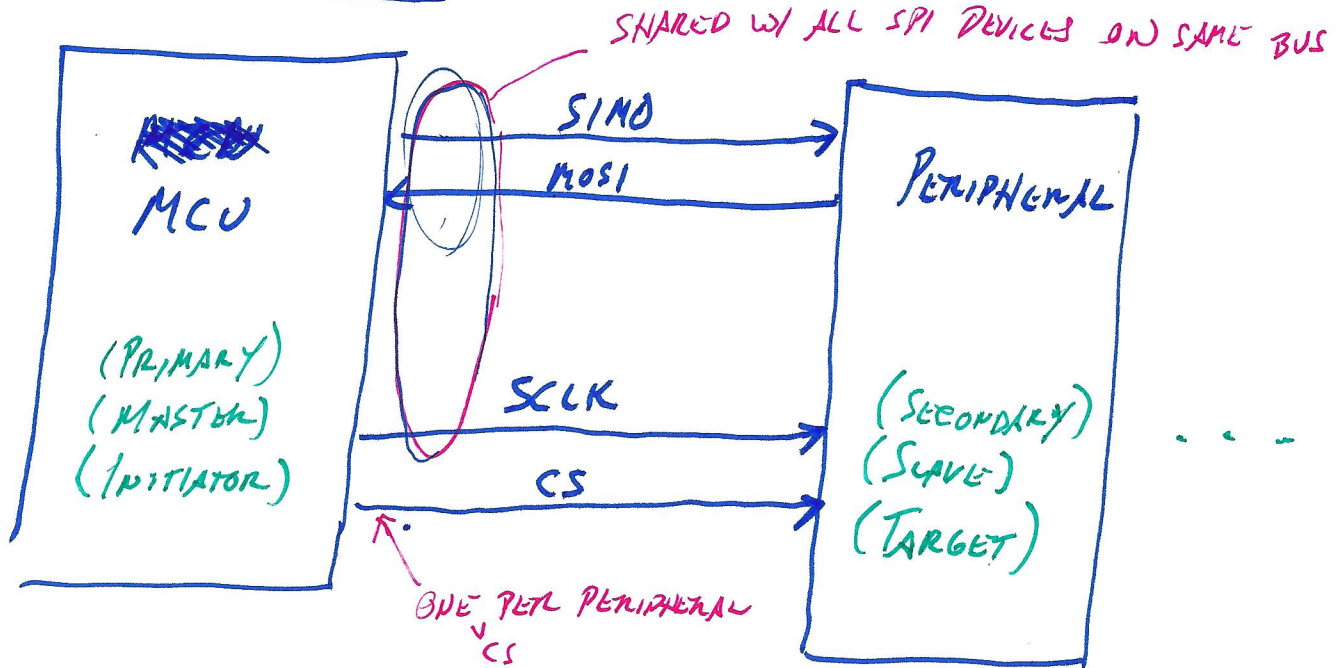
Press to Save



Serial Peripheral Interface Bus (SPI)

>> Used primarily for synchronous serial communications between a CPU and peripherals "within the box"

-- Synchronous = shared clock (supplied by master device)



>> Usually a 4 wire connection (sometimes 3-wire)

SIMO = Slave In/Master Out data line

SOMI = Slave Out/ Master In data

SCLK = Serial Clock (Called UCAxCLK in MSP430 Documentation)

CS = Chip Select

SELECTS WHICH DEVICE IS ACTIVE.

>> SPI is somewhat loose standard --> Different from I²C

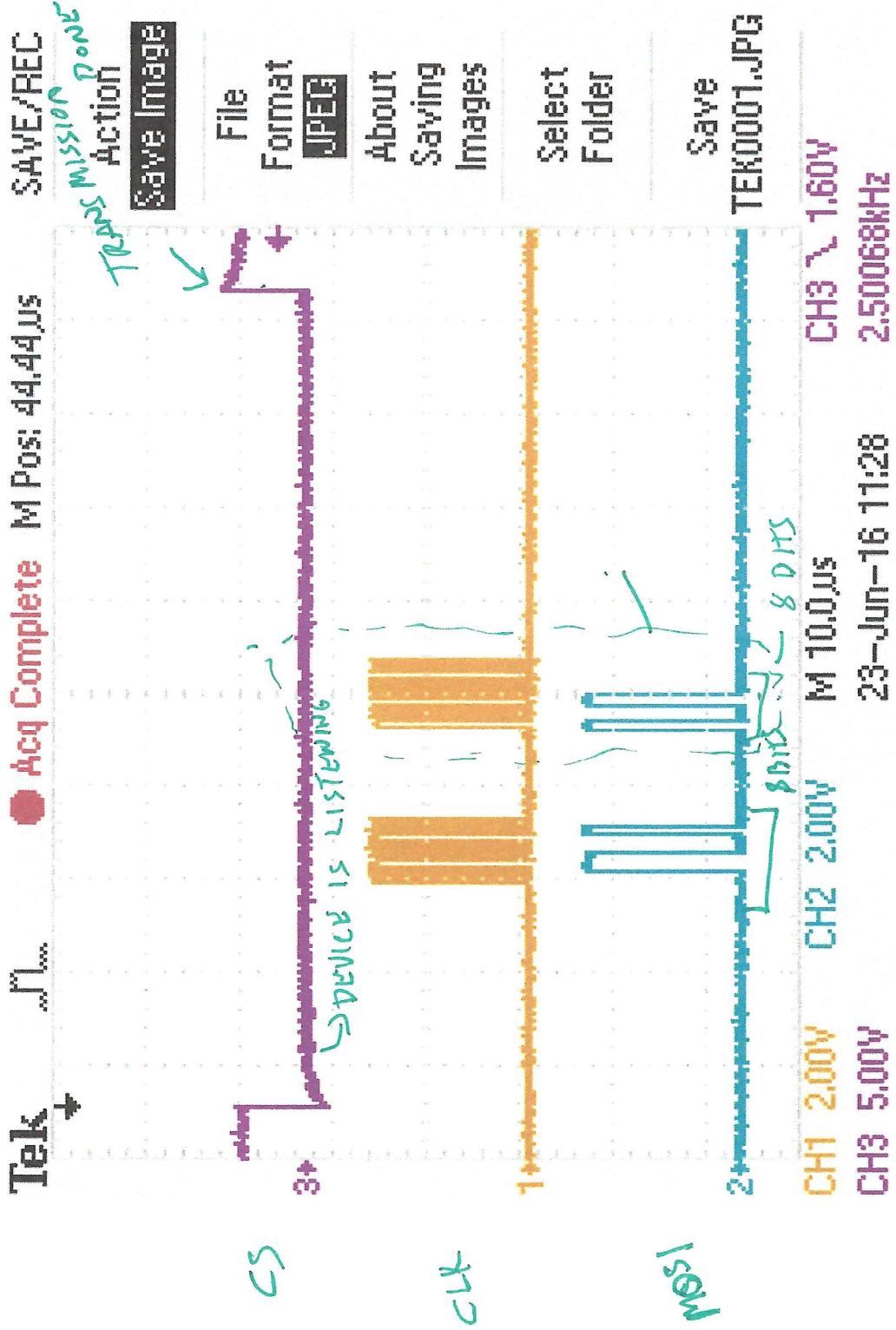
DON'T NEED SPECIFIC BAUD RATE OR CLOCK FREQ.

How will you know what to use?

--> SPI, I²C, asynchronous serial (RS-232), other?

Sensors or other peripheral devices will specify the interfaces with which they are compatible

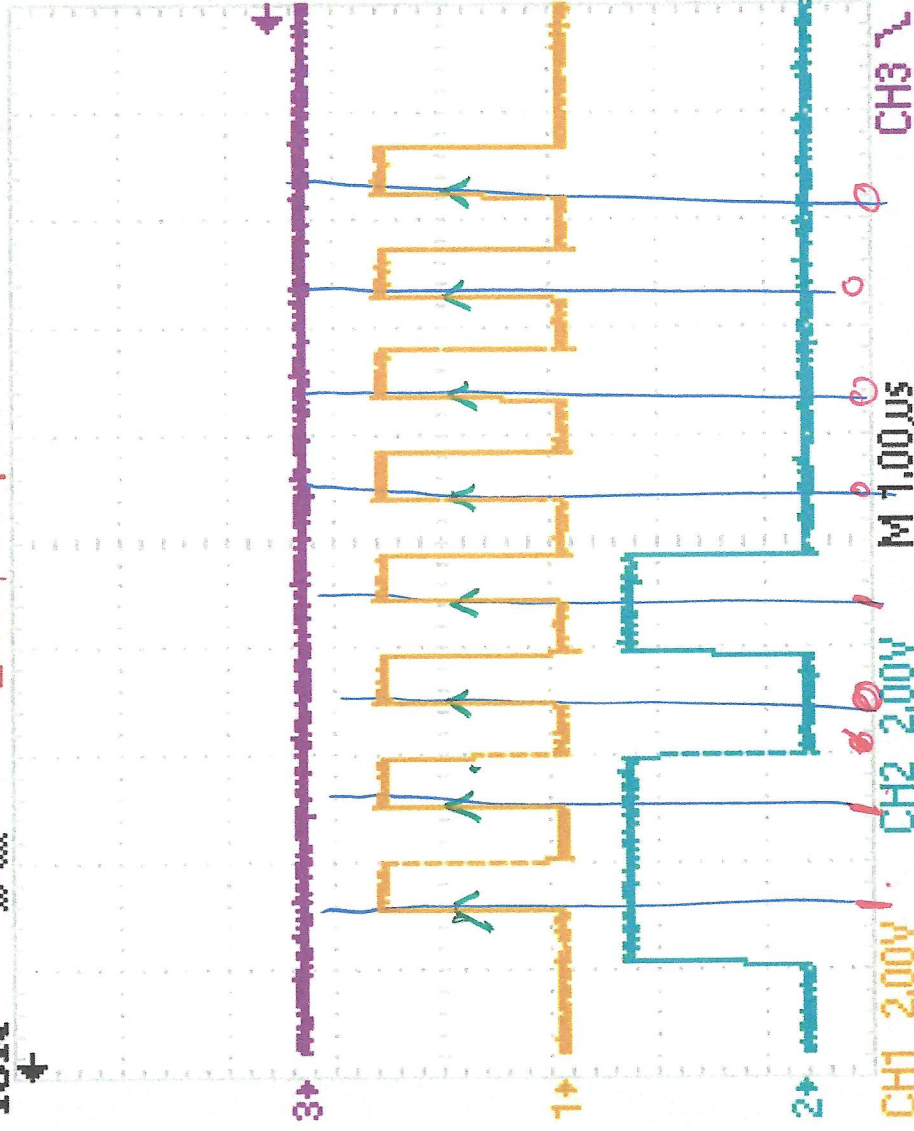
EXAMPLE SPI TRANSMISSION



SAVE/REC

Acq Complete M Pos: 44.44µs

Tek



Action
Save Image

File
Format
JPEG

About
Saving
Images

Select
Folder

Save
TEK0001.JPG

CS

JK

MS1

DATA: 1101 0000

[D CH]

NEED

- MSB FIRST (USUALLY)
- READ ON RISING OR FALLING EDGE

NO. 7

Example Data format: MCP4921 Digital to Analog Converter (DAC)

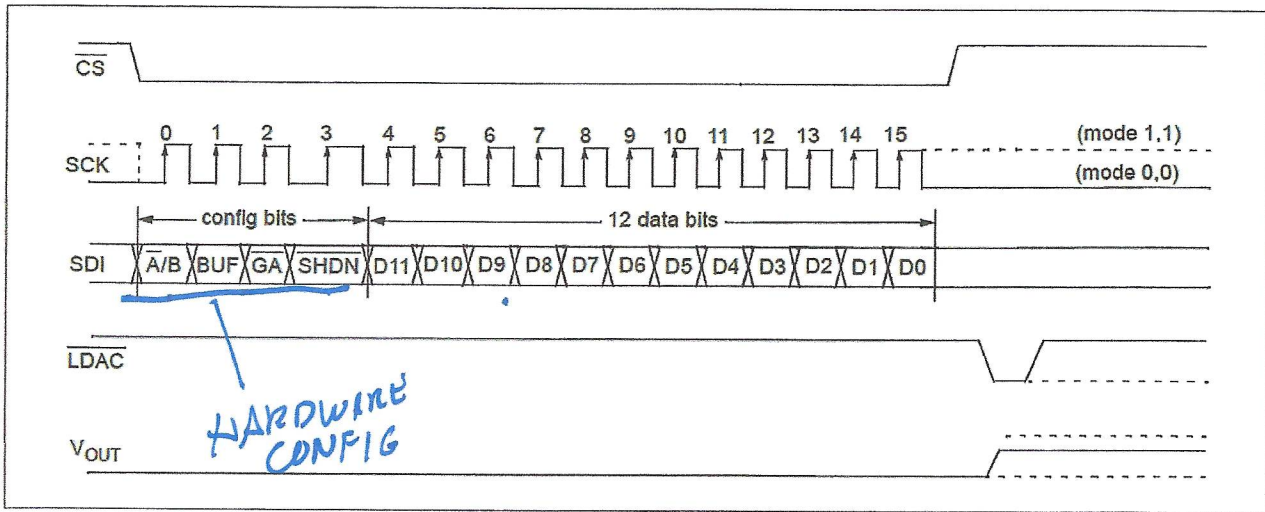


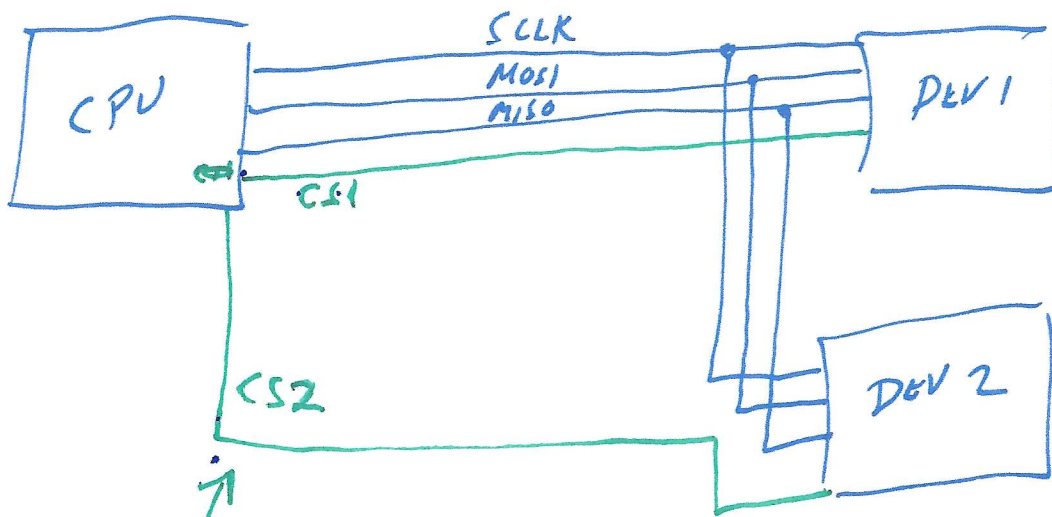
FIGURE 5-1: Write Command.

HERE, DAC EXPECTS 16 BITS:

- BITS 15-12: HARDWARE CONFIG.

- BITS 11-0: DAC CODE.

CONNECTING MULTIPLE DEVICES TO SPI BUS:



EVERY DEVICE GETS A CS.

Example SPI Peripherals

Our lab boards have 2 peripherals that are SPI devices, the LCD screen and the digital-to-analog converter (DAC). However, we could readily connect others as the USCI pins and some digital IO pins are available through the headers

Sharp 96x96 LCD Display

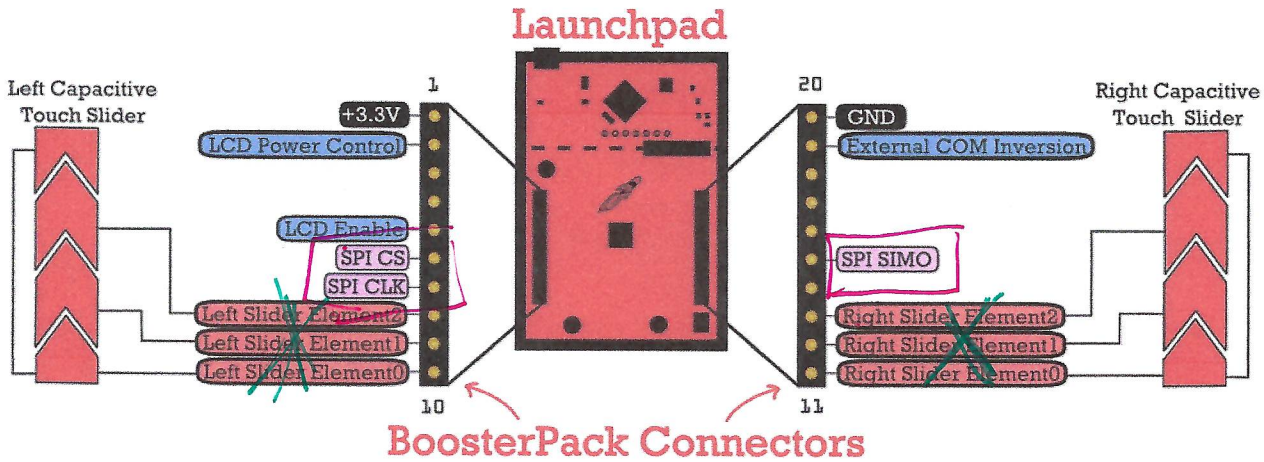


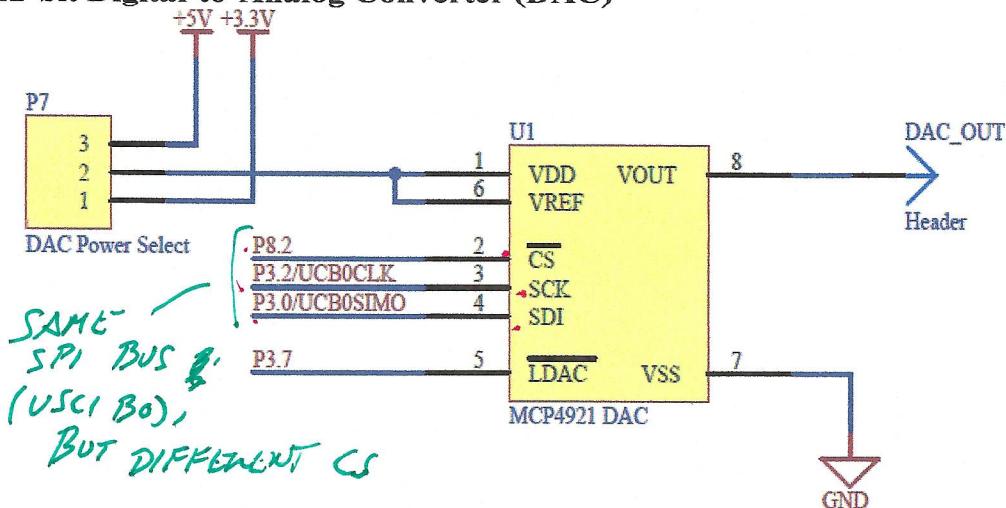
Figure 3. BoosterPack Default Pinout

Interface Pins

- SPI CS: P6.6 (Digital I/O)
- SPI CLK: P3.2 (Function mode, UCB0CLK)
- SPI SIMO: P3.0 (Function mode, UCB0SIMO)
- Two additional digital I/O pins to provide power and enable LCD (P6.5 and P1.6)

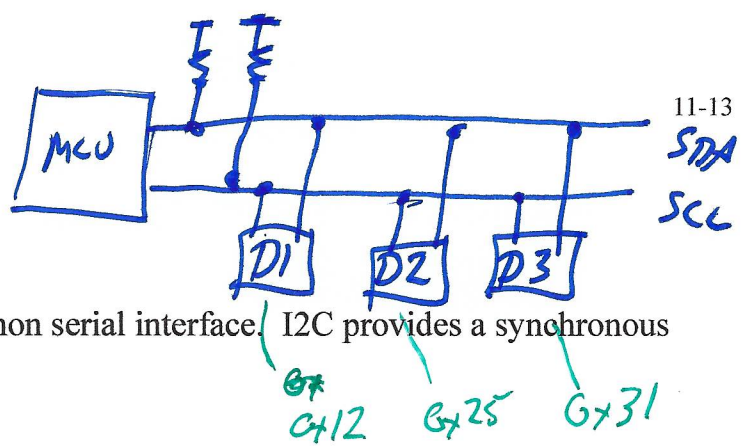
Note: LCD does not send data to the MSP430, so the SOMI line is not used!

MCP4921 12-bit Digital-to-Analog Converter (DAC)



Digital to Analog Converter: Converts 12-bit digital code into an analog voltage (in some range V_{Ref+} to V_{Ref-}). Sound familiar?
Can use this to generate analog signals!

Inter-Integrated Circuit (I²C)



I²C (also, I2C) is another increasingly common serial interface. I2C provides a synchronous interface using only two wires!

An I2C interface is comprised of two wires, called the “I2C bus”:

- SCL: Serial clock
- SDA: Serial data

I2C bus fundamentals:

- All peripherals are connected to the same two wires
- Both SCL and SDA require *pull-up* resistors such that both lines default to logic high.
- Both lines are *bidirectional*
- Speeds are standardized: typically 100kbps (standard mode) or 400kbps (fast mode)

Unlike SPI, I2C is more rigidly standardized and has strict timing requirements on how the two lines can be used. There are a few standard operations:

- START
- STOP
- ACK: Acknowledge transmission
- NACK: Negative acknowledgement

Every device datasheet will tell you how and when it expects to receive these commands.

How are devices selected?

With only two wires, we have nothing like a chip select (CS) to wake up individual devices! In I2C, every device has an *address*, which is often programmed at the factory.

Before any transmission, the master sends the address of the device it wants to use—the device should only respond to the request if it has a matching address.