Overview

- This lecture
  - A/D and D/A devices and their operations
    - ADC unit of dsPic33
    - MCP4822 (mounted on FlexUI board)

  The content of this lecture is partially covered by the lab manual (pp. 14-16, 29-32, and pp38-40). Additional material is on dsPic data sheet (section 16 (ADC)), MCP4822 data sheet

A/D D/A dsPic board

- dsPic board
  - Analog to Digital Converter (ADC)
    - 10-bit or 12-bit resolution
    - 0 to +3.3 volt range is the current configuration
  - Digital to Analog Converter (DAC)
    - 12-bit resolution
    - 0 to +4.096 volt range

dsPic Board ADC

- 2 Analog to Digital converters are built into the dsPic33 microcontroller. They can have 10-bit or 12-bit resolution.

- ADC1 has access to analog input pins AN0-31 and ADC2 has access to analog input pins 0-15. ADC input pins are shared with digital I/O pins.

- Four channels are connected as such

PIN	**PORT**	**ADC Channel**	**Connection**
RB4	B	CH4	Joystick's x-axis
RB5	B	CH5	Joystick's y-axis
RB15	B	CH15	Balance Board's x-axis
RB9	B	CH9	Balance Board's y-axis

dsPic ADC operation overview

- Operation overview (details in the lab manual and dsPic data sheet)
  - Initialization
  - Select analog input pin (AN0-31)
  - Start conversion
  - Wait for conversion to finish
  - Collect conversion result

dsPic ADC Initialization

- The following steps must be taken to configure the ADC for use:
  - disable ADC by clearing the ADxCON1 ADON bit.
  - set the appropriate pin to input using TRISx, where x is the appropriate port (A, B, C, etc.)
  - set the appropriate pin to analog using the ADxPCFGL for AN0-15 or ADxPCFGH for AN16-31.
  - set ADxCON1 to configure 10 or 12 bit Operation Mode, Data Output Format, and Sample Clock Source. We suggest using the appropriate bit mode, integer, and automatic conversion
  - set ADxCON2 to configure scanning sampling. We suggest you set it to zero (i.e. no scanning sampling).
  - set ADxCON3 to configure the Conversion Clock Source, Auto Sample Time Bits, and Auto Conversion Clock Select (See sample code and the ADC reference manual for proper lab values).
  - Enable the ADC unit, using ADxCON1 by setting the ADON bit.
- Note that each analog pin ANx is shared in the dsPic with a bit of a digital I/O port (see pag.7 data-sheet of dsPic).

dsPic AD: coding example for initialization

- The code below sets ADC1 to use analog input pin AN20 (PIN = RE8), 10 bit resolution, integer output, automatic conversion mode, with no scanning.

```
//disable ADC
CLEARBIT(AD1CON1bits.ADON);
//initialize PIN
SETBIT(TRISEbits.RE8);                  //set TRISE RE8 to input
CLEARBIT(AD1PCFGH.PCFG20);              //set ADC1 pin AN20 as analog
//Configure AD1CON1
CLEARBIT(AD1CON1bits.AD12B)             //set 10b resolution
AD1CON1bits.FORM = 0;                   //set integer output
AD1CON1bits.SSRC = 0x7;                 //set automatic conversion
//Configure AD1CON2
AD1CON2 = 0;                            //not using scanning sampling
//Configure AD1CON3
CLEARBIT(AD1CON3bits.ADRC);             //internal clock source
AD1CON3bits.SAMC = 0x1F;                //sample-to-conversion clock = 31Tad
AD1CON3bits.ADCS = 0x2;                 //Tad = 3Tcy (Time cycles)
//Leave AD1CON4 at its default value
//enable ADC
SETBIT(AD1CON1bits.ADON);
```

dsPic ADC Sampling&Conversion

- The following steps must be taken to sample an ADC channel:

  - set ADxCHS0 to the proper analog input pin (AN0-AN31),
  - set the SAMP bit in the ADxCON1 to start a sample.
  - wait on DONE bit in the ADxCON1 to signal the sampling and conversion are done
  - clear DONE bit and retrieve the value from ADCxBUF0

dsPic: coding example for ADC operation

- The code below will sample ADC1 analog input pin 20 with the previous configuration

```
AD1CHS0bits.CHS0SA = 0x014;      //set ADC1 to sample AN20
SETBIT(AD1CON1bits.SAMP);        //start to sample
while(!AD1CON1bits.DONE);        //wait for conversion to finish
CLEARBIT(ADCON1bits.DONE);       //MUST HAVE! clear conversion done bit
return ADC1BUF0;                 //return sample
```

The General Conversion Rule

- Convert the analog voltages to a N-bit digital representation is a special case of measurement conversion.

- Let's take the range between water freezing and boiling points at sea level. The measure using Celsius is from 0 to 100 while the measure using Fahrenheit is from 32 to 212.

- How can we convert from Celsius to Fahrenheit?
- Hint: **The general conversion rule is derived from the observation that two different linear measures that measure the same physical quantity must be proportional to each other**. For example, 10% in range_1 maps to 10% in range_2, and 70% in the range_1 maps to 70% in range_2.

- Quiz 1: How to convert 60c to Fahrenheit?

The General Conversion Rule

- Convert the analog voltages to a N-bit digital representation is a special case of measurement conversion.

- Let's take the range between water freezing and boiling points at sea level. The measure using Celsius is from 0 to 100 while the measure using Fahrenheit is from 32 to 212.

- How can we convert from Celsius to Fahrenheit?
- Hint: The general conversion rule is derived from the observation that two different linear measures that measure the same physical quantity must be proportional to each other. For example, 10% in range_1 maps to 10% in range_2, and 70% in the range_1 maps to 70% in range_2.

- Quiz 1: How to convert 60c to Fahrenheit?
- Answer:
  - 60c is equivalent to 60% in range of Celsius;
  - 60% in the range of Fahrenheit is: (212 - 32) * 60 / 100 + 32 = 140 F

The General Conversion Rule

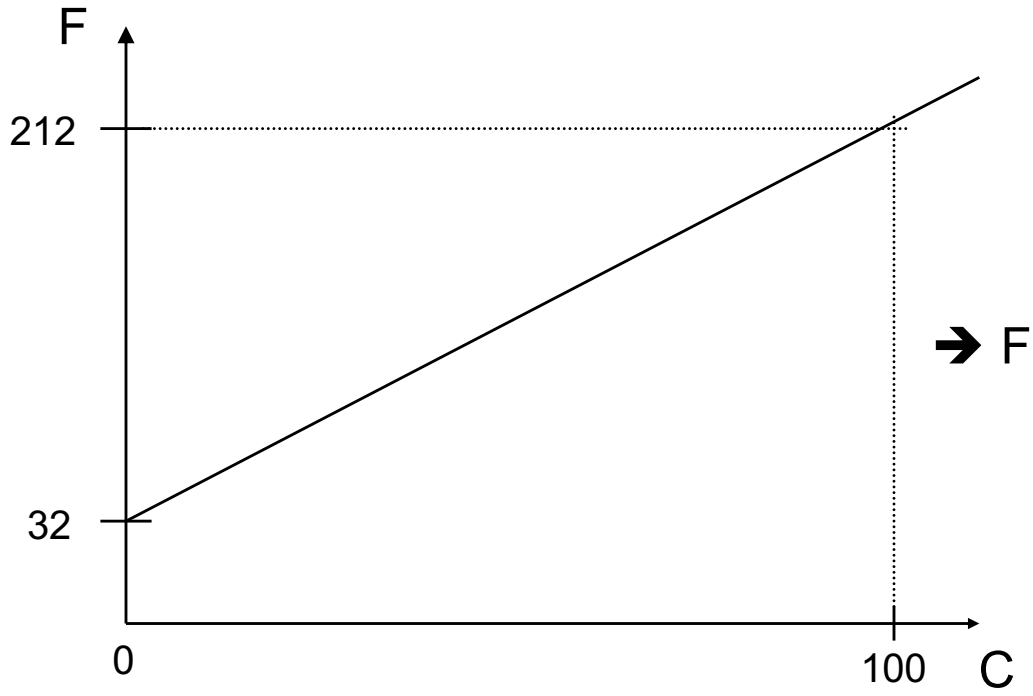
- Convert the analog voltages to a N-bit digital representation is a special case of measurement conversion.

- Let's take the range between water freezing and boiling points at sea level. The measure using Celsius is from 0 to 100 while the measure using Fahrenheit is from 32 to 212.

- How can we convert from Celsius to Fahrenheit?
- Hint: The general conversion rule is derived from the observation that two different linear measures that measure the same physical quantity must be proportional to each other. For example, 10% in range_1 maps to 10% in range_2, and 70% in the range_1 maps to 70% in range_2.

- Quiz 2: Can you generalize your result and find a graphical representation of the general formula to convert from measure 1 to measure 2?

The General Conversion Rule

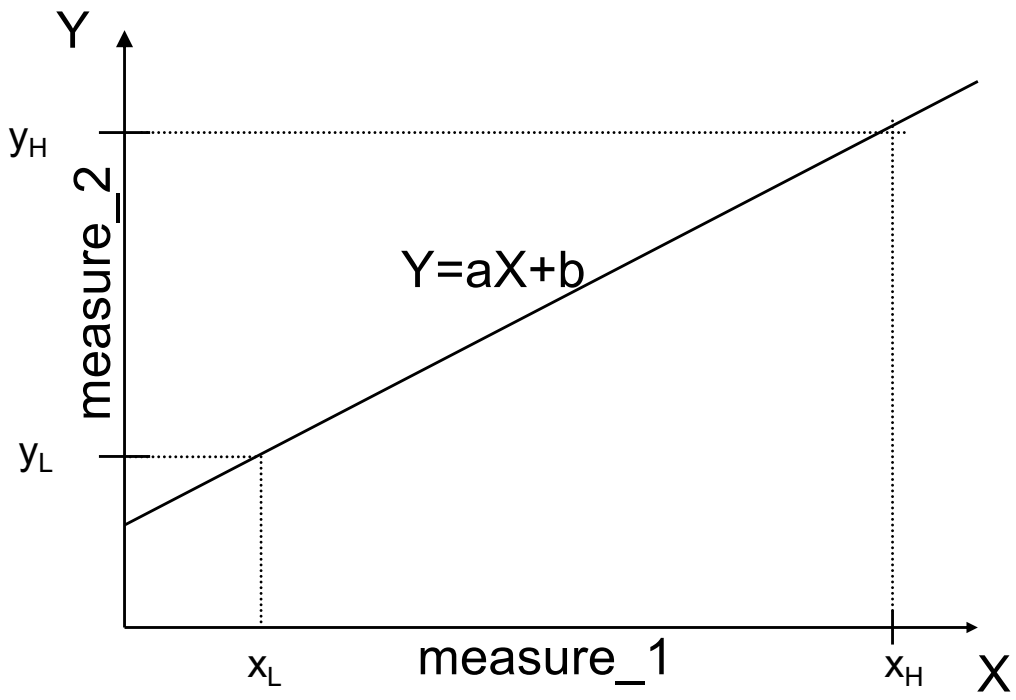


Y=aX+b

➔ F = C (212 – 32)/100 + 32

- Quiz 2: Can you generalize your result and find a graphical representation of the general formula to convert from measure 1 to measure 2?
- Answer: we can graphically represent the conversion between Celsius and Fahrenheit by using the equation of a straight line (see picture above!). We can further generalize our formula for conversions between any two different linear measures that measure the same physical quantity (see next page).

Appendix: The General Conversion Rule



$$\begin{cases} y_L = ax_L + b \\ y_H = ax_H + b \end{cases}$$

$$y_H - y_L = a(x_H - x_L) \quad \Rightarrow \quad a = \frac{y_H - y_L}{x_H - x_L}$$

$$b = y_L - ax_L = y_L - \frac{y_H - y_L}{x_H - x_L} x_L$$

$$Y = \left(\frac{y_H - y_L}{x_H - x_L} \right) X - \left(\frac{y_H - y_L}{x_H - x_L} \right) x_L + y_L$$

$$Y = \left(\frac{y_H - y_L}{x_H - x_L} \right) \left(X - x_L \right) + y_L$$

Answer: measure_2 = ((measure_1 - Measure_1_Lower_Limit)/range_1)*range_2 + Measure_2_lower_limit.

A/D and D/A Mapping

A/D: ((_____ - _____)/(_____))*(_____) + _____

((V - V_lower_limit)/(V_range))*(digital_range) (+ digital_lower_limit = 0)

D/A: ((_____ - _____)/(_____))*(_____) + _____

((digital_level)/digital_range)*analog_range + analog_lower_limit

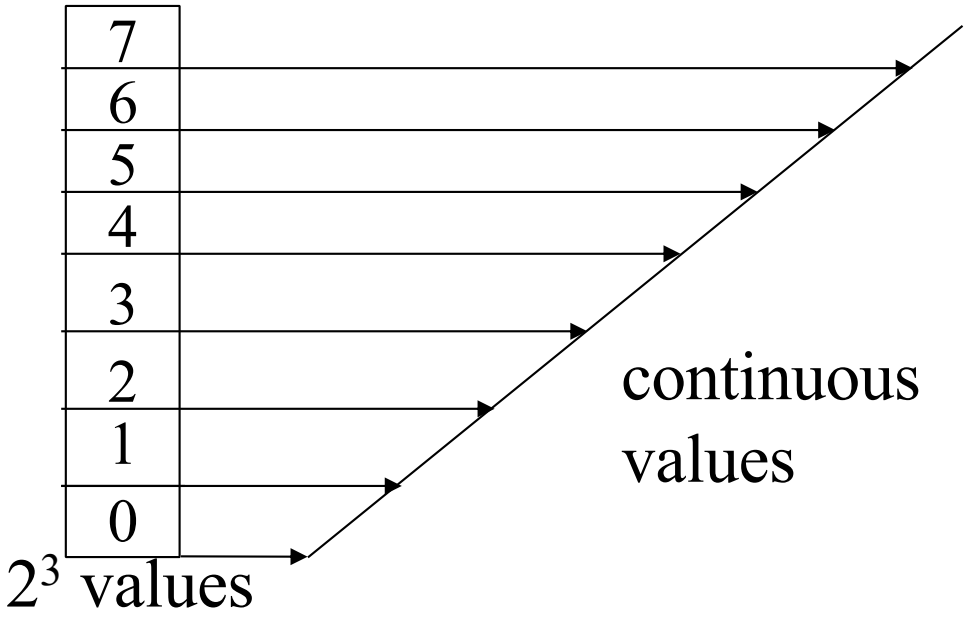
Example: analog voltages range is -1 to +2v and the digital_levels are from 0 to $2^n - 1$.
If $2^n - 1 = 3$, the digital output is 2 bits with a range of 3 - 0 = 3.

A/D: V = +1: ((1 - (-1))/3)*3 + 0 = 2. That is, 10 binary: 10B

D/A: 10B: (2/3)*3 + (-1) = +1v

Quantization

- Quantization is the taking of discrete samples from continuous values for some physical quantities.
- When we sample by using the ADC, we are mapping the continuous voltage range to a fixed number of digital values.



Quantization Example

- ADC samples between -3 to +4 volts and rounds to a 3-bit number.
- Reading an analog voltage value of +2.9V results in ~ 6 in 3 bit representation.
- Converting back to analog: Resulting in **quantization error** of 0.1V

$$\left(2.9-\left(-3\right)\right)*\frac{\left(2^3-1\right)-0}{4-\left(-3\right)}=5.9\sim 6$$

$$6*\frac{4-(-3)}{(2^3-1)-0}+(-3)=3.0$$

- A/D: ((V - V_lower_limit)/(V_range))*(digital_range) (+ digital_lower_limit = 0)
- D/A: ((digital_level)/digital_range)*analog_range + analog_lower_limit

How Does This Affect Me?

- Video: 24 vs. 16 vs. 8-bit color
- Music: 24 vs. 16 vs. 8-bit sound

- DAS1600 ADC: 12-bit resolution (affects precision when reading an analog input)

- The N = 12 bits can be configured to represent different analog voltage ranges,
  - e.g., -2.5 V to +2.5 V, -5 V to +5 V and -10 V to +10 V.

- Quiz: What are these ranges for? What should be the rule in picking a range?

How Does This Affect Me?

- Video: 24 vs. 16 vs. 8-bit color
- Music: 24 vs. 16 vs. 8-bit sound

- DAS1600 ADC: 12-bit resolution (affects precision when reading an analog input)

- The N = 12 bits can be configured to represent different analog voltage ranges,
  - e.g., -2.5 V to +2.5 V, -5 V to +5 V and -10 V to +10 V.

- Quiz: What are these ranges for? What should be the rule in picking a range?
- Answer: These ranges represent the max and min permitted voltages for the ADC. Pick the smallest range that still includes the max and min voltage values that can be read from the source.

Analog input calibration

- Readings of a potentiometer can vary from one experiment to another. A program should always start by calibrating the analog input before using it.

- Due to the construction of the joysticks, the voltage range seen on each analog axis will differ. If an accurate position is required, obtain calibration data (such as the x-axis and y-axis voltages at the upper left and lower right corners) and scale future samples accordingly. Each pair (position, read_digital_value) represents calibration data: you need at least two pairs to perform calibration.

FlexUI Board DAC

- Digital to analog converter chip (MCP4822). This chip has 12 bits of resolution over the range zero to 4.096 volts.

- The chip provides a serial interface with pin SDI. 16 bits must be “clocked” in one at the time starting with the most significant bit.
  - Bit 15 is the Channel select (A=0/B=1)
  - Bit 14 is not used (set to 0 or 1)
  - Bit 13 is gain (set to 1 for max Vout=4.096)
  - Bit 12 is Output Shutdown Control (set to 1 for “active mode operation”; Vout is available)
  - Bits 11-0 is the 12 bit data

FlexUI Board DAC

- Registers that allow to control the DAC chip on FlexUI board:

Microcontroller	DAC	Description
PORTD bit 8	$\overline{\text{CS}}$	Chip Select
PORTB bit 11	SCK	Serial Clock Input
PORTB bit 10	SDI	Serial Data Input
PORTB bit 13	$\overline{\text{LDAC}}$	Latch DAC Input

Table 7: Digital to analog converter chip connections to the microcontroller

FlexUI Board DAC operation

- Operation overview
  - Configure the DAC's pins on the dsPic
  - Set the /CS pin low to enable communication
  - Send a bit (repeat 16 times)
    - Set the SCK pin low
    - Set the SDI pin to the bit you want to send
    - Set the SCK pin high
  - Once the entire 16 bit value has been sent, set the /CS pin high
  - Toggle the /LDAC pin low then high to actually output the new voltage.

FlexUI Board DAC programming

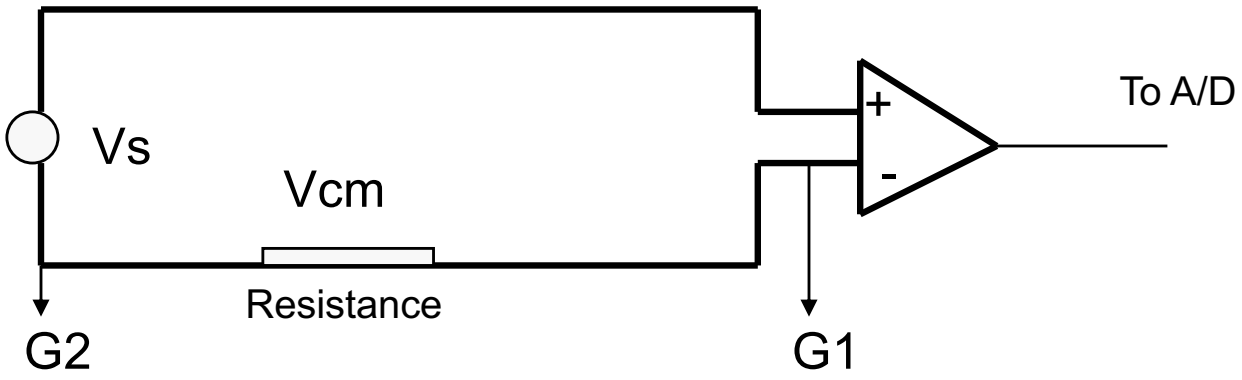
- The DAC SCK line is connected to PORTB bit 11.
- The DAC SDI line is connected to PORTB bit 10.

- How to set and clear the SCK line.
  - PORTB |= BV(11); // Set SCK high
  - PORTB &= ~BV(11); // Set SCK low

- To set the SDI line equal to bit 2 in a variable var, use the following.
  - PORTB &= ~BV(10); // Set SDI low
  - PORTB |= (var & BV(2)) >> 2 << 10;
    // Set SDI high only if bit 2 in var is 1.

Single Ended Input

- External voltage can be connected with A/D cards via single ended inputs or differential inputs.
- Single ended input measures the difference between ground and the input signal. It is susceptible to EMI interference and voltage difference between grounds of the A/D card and the ground of the signal source. They are fine in a low noise lab environment.



Differential Inputs

- Differential connections are insensitive (e.g. up to 10 v) to ground differences and EMI.
- However, electronically differential connections require twice input lines. For example, DAS 1600 may have 16 single ended A/D inputs or 8 differential A/D inputs.