MACQUARIE University
SYDNEY·AUSTRALIA

Interfacing with an analog world

ELEC3042 EMBEDDED SYSTEMS

Lecture 5

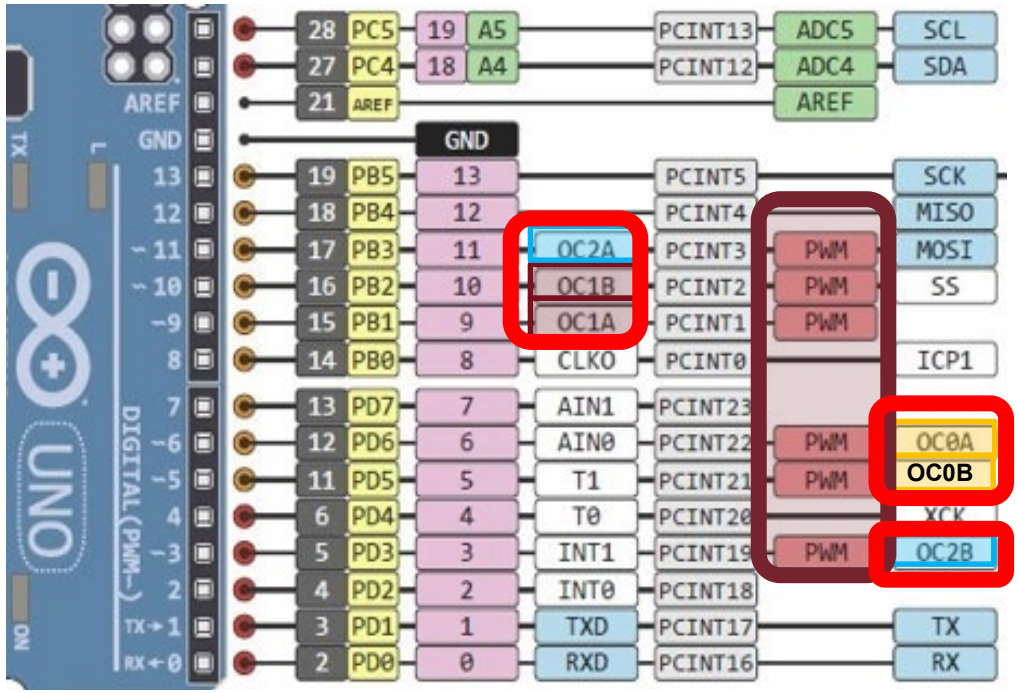
Digital vs Analog

MACQUARIE University
SYDNEY·AUSTRALIA

- Real world is analog (continuous)
- CPUs are digital
- For a computer to interact with the real world, we need
  - Digital to Analog (D-to-A) conversion – convert digital signal to analog
  - Analog to Digital (A-to-D) conversion – convert analog signal to digital

Using Timers to create analog output

- Timers can generate square waveforms that are outputted on pins



Waveform generation

- Change pin output value on compare match

CTC

Table 16-1. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match.
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level).
1	1	Set OC1A/OC1B on Compare Match (Set output to high level).

PWM

Table 16-2. Compare Output Mode, Fast PWM(1)

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

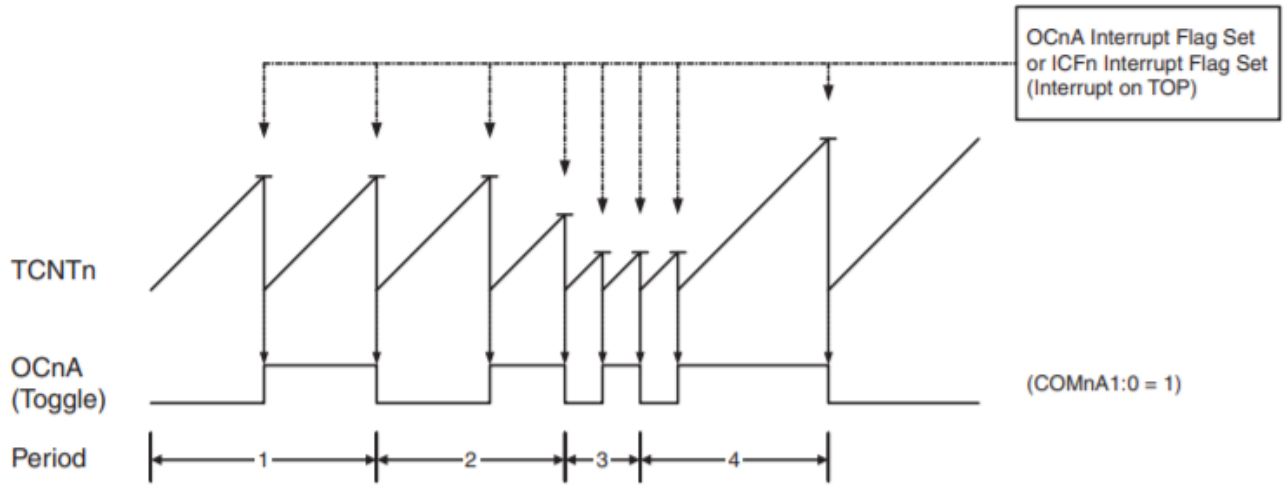
Table 16-3. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM(1)

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 11: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match when up-counting. Set OC1A/OC1B on Compare Match when downcounting.
1	1	Set OC1A/OC1B on Compare Match when up-counting. Clear OC1A/OC1B on Compare Match when downcounting.

CTC

- Single-slope operation: TCNTn counts from BOTTOM to TOP and resets to BOTTOM
- Toggle output when count reaches value in OCnA & reset TCNTn
- Set/change period of output waveform by setting/changing OCnA value

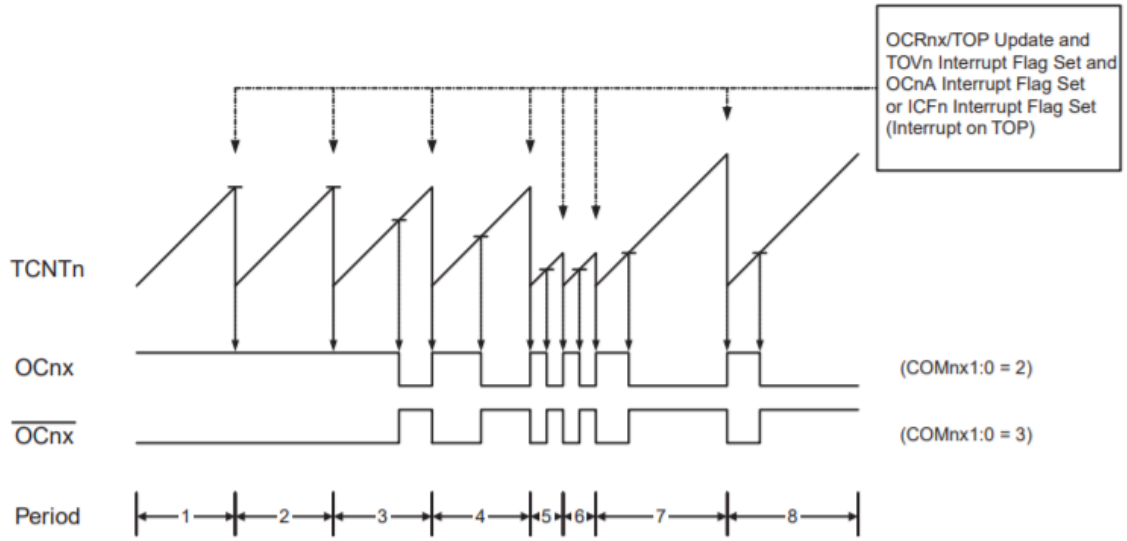
Figure 16-6. CTC Mode, Timing Diagram



Fast PWM

- Single-slope operation
- Non-inverting mode (inverting mode is opposite):
  - output low when count reaches compare value
  - output high when count reaches TOP value
- Change compare value to change duty cycle

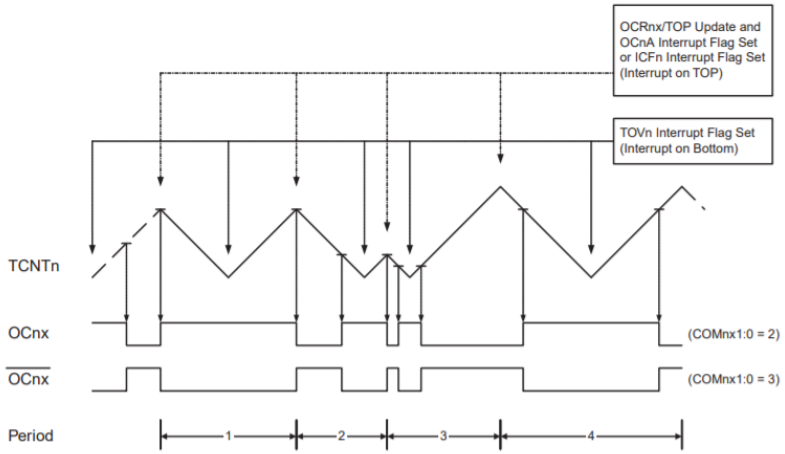
Figure 16-7. Fast PWM Mode, Timing Diagram



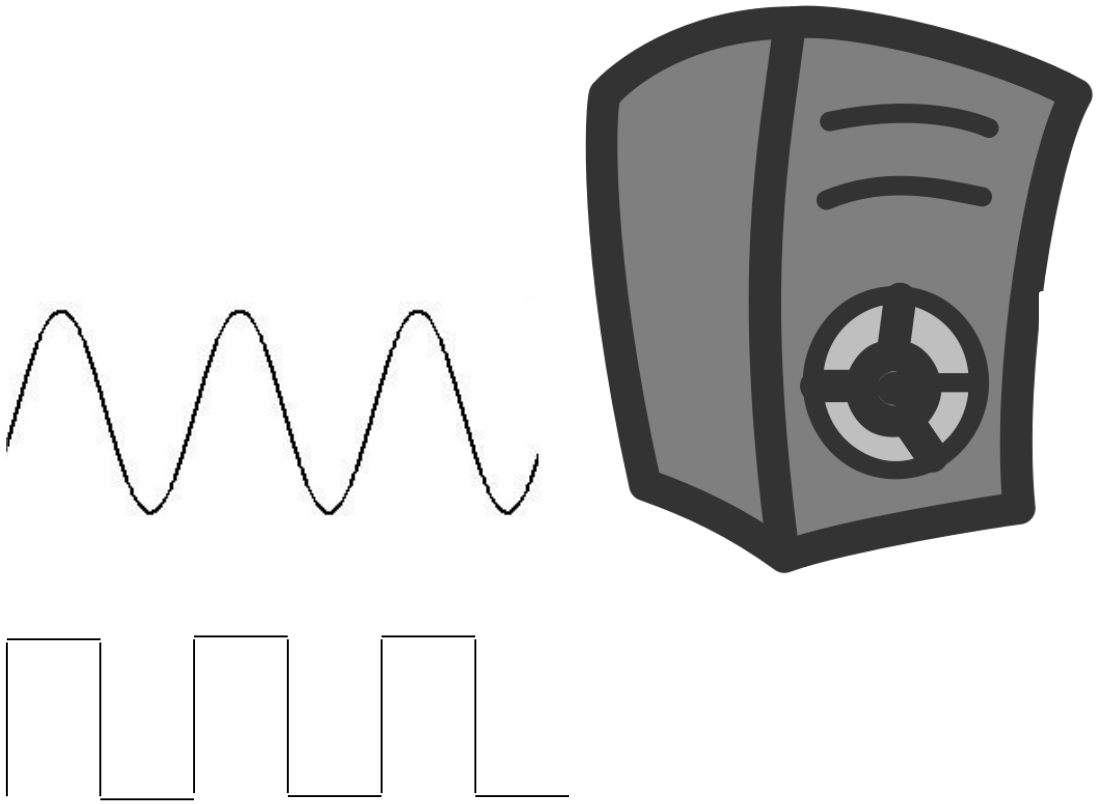
Phase correct, Phase & Frequency correct PWM

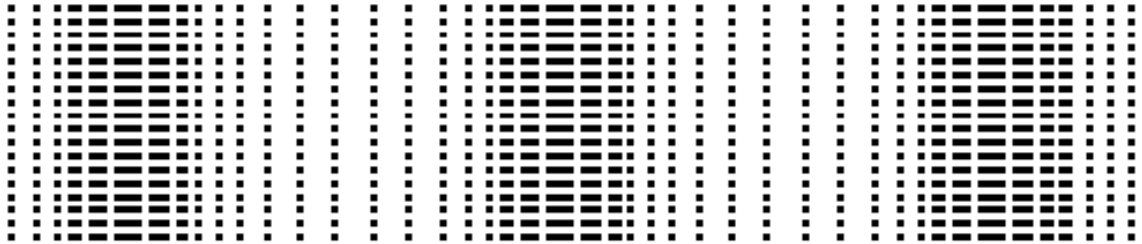
- Dual-slope operation: TCNTn counts from BOTTOM to TOP and down to BOTTOM
- Non-inverting mode (inverting mode is opposite):
  - Output low on compare match while upcounting
  - Output high on compare match while downcounting
- Lower max frequency than FAST PWM but pulses are symmetric
- Difference between phase correct, and phase & frequency correct is when OCRnx register is updated

Figure 16-8. Phase Correct PWM Mode, Timing Diagram



Making a 440 Hz sound using CTC





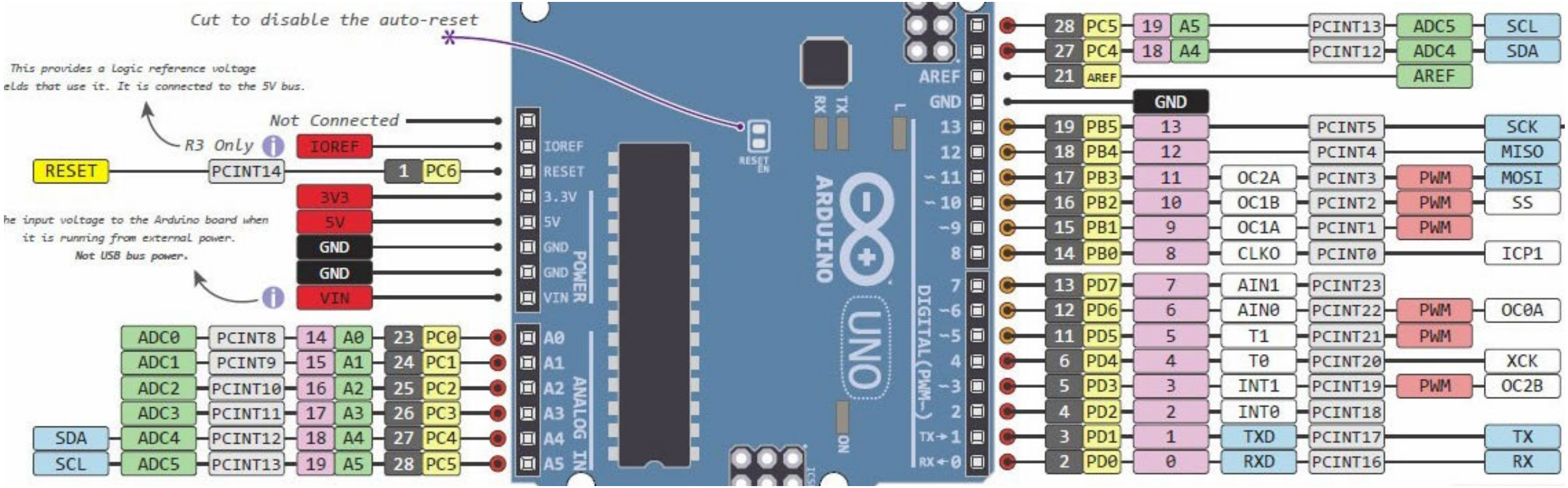
Choose a timer

MACQUARIE University
SYDNEY·AUSTRALIA

OCRnA = clock frequency / (2 * desired frequency)

Prescaler	Apparent Clock Frequency	# clock ticks for 440 Hz	OCCRnA value for 440 Hz	Error (Hz)
No scaler	16 MHz	18181.82	18181	0.0044
/8	2 MHz	2272.73	2272	0.0528
/64	250 kHz	284.09	283	-0.1408
/256	62 500 Hz	71.02	70	-0.1408
/1024	15 625 Hz	17.76	17	5.9722

Which Pin?



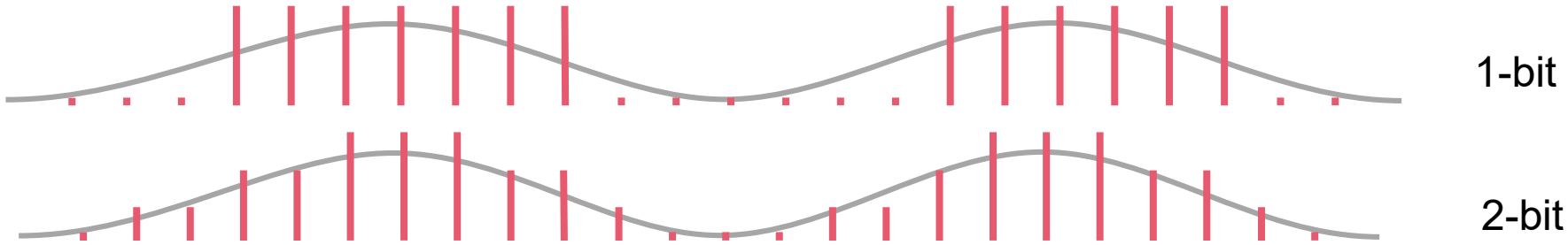
Exercise

MACQUARIE University
SYDNEY·AUSTRALIA

Write the setup code to generate a 440 Hz sound

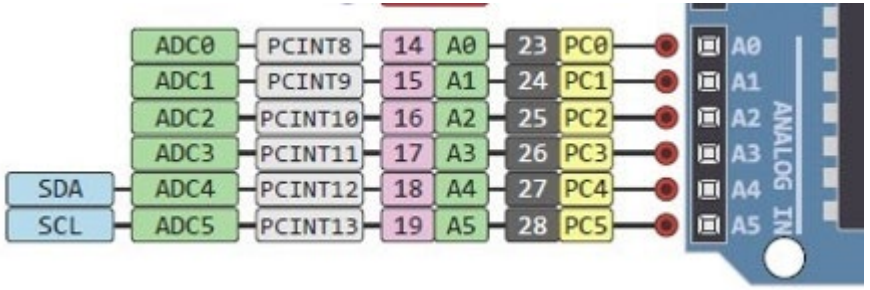
ADC: Sampling & Resolution

- Conversion to digital occurs at discrete moments in time
- Sampling rate (number of conversions per second) is limited by conversion time
- Sampling rate limits the bandwidth of the digital representation (Nyquist frequency)
- Resolution is the number discrete levels available to represent the amplitude of a signal and determined by the number of bits of ADC



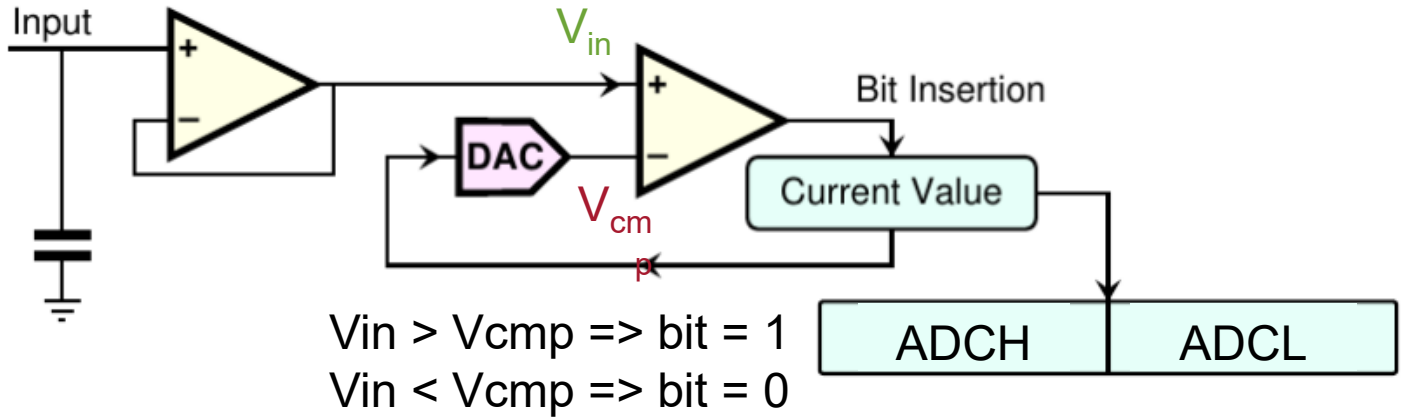
Analog-to-Digital Convertor on ATmega328P

- One 10-bit ADC
- Multiplexed across the PORTC pins
  - That is, if 2 pins are using the ADC, the total sampling rate will be divided across the 2 pins
- Max sampling rate is dependent on resolution and clock rate
  - Normal clock frequency 50kHz to 200kHz
  - Accepting lower resolution allows >200 kHz clock => higher sampling rate
- ADC has prescaler to generate an acceptable clock frequency
  - Set by ADPS bits in ADCSRA

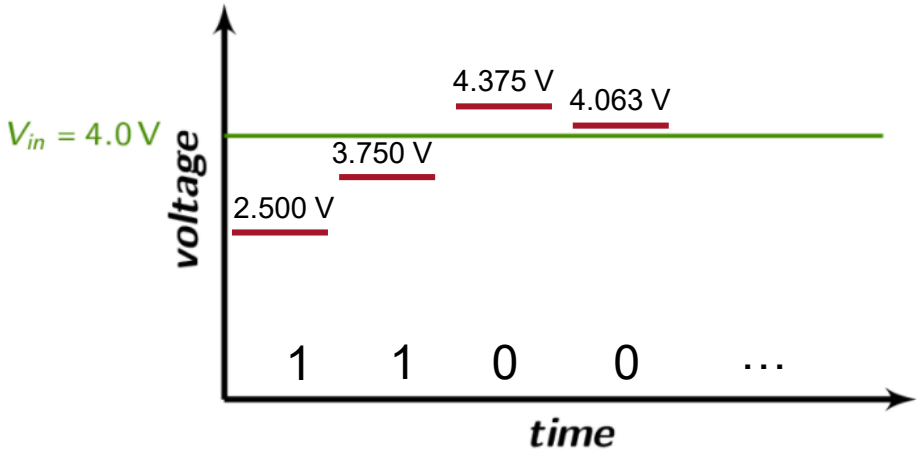


Successive Approximation

- Samples input signal voltage
- Input signal voltage is held constant and successively compared to a comparison voltage to find the binary representation one bit at a time starting from MSB
- Conversion normally takes 13 ADC clock cycles
  - First conversion takes 25 ADC clock cycles because the analog circuitry needs some time to initialise



Vin > Vcmp => bit = 1
Vin < Vcmp => bit = 0



$5V \times 1/2^1 = 2.500\ V$
$5V \times 1/2^2 = 1.250\ V$
$5V \times 1/2^3 = 0.625\ V$
$5V \times 1/2^4 = 0.313\ V$
…
$5V \times 1/2^{10} = 0.005\ V$

Conversion Result

- When conversion is complete, result is stored in ADC register and ADIF flag is set
- ADC is made up of two 8-bit registers: ADCH & ADCL
- To get 16-bit result, read ADC (or ADCL first, then ADCH)
- Result can be left or right-adjusted (depends on ADLAR bit)
- If 8-bit (or less) resolution is required, use left-adjust and read ADCH only

24.9.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	

Right-adjusted (default)

24.9.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

Left-adjusted

Triggering a conversion

MACQUARIE University
SYDNEY·AUSTRALIA

- Manually
  - Write 1 to ADSC
- Free running
  - Starts a new conversion as soon as the ongoing one is finished
- Using an automatic trigger
  - Allows conversions at a fixed interval

Table 24-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Setting up ADC for conversion

MACQUARIE University
SYDNEY · AUSTRALIA

- Select PORTC pin(s) for ADC and configure pin(s) as an input (no pullup resistor)
- Configure A-to-D module
  - ADMUX register
    - Select Voltage reference source (AVcc)
    - Set left or right adjust output (ADLAR bit)
    - Set MUX bits for a pin to get analog value
  - ADCSRA/ADCSRB registers
    - Enable ADC (ADEN bit)
    - Enable interrupt (if desired)
    - Set an appropriate clock prescaler
    - Set desired conversion trigger
  - DIDR0 register
    - Disable digital input of ADC pins to conserve power
- Do an initial conversion (because first one takes 25 ADC clock cycles)
    - Set ADSC bit to 1 in ADCSRA register

Exercise

MACQUARIE University
SYDNEY·AUSTRALIA

Write the setup code to do an 8-bit analog-to-digital conversion on PC0. Use AVcc for reference and clock prescaler = 128. Set up the system so that it will be free running with no interrupts generated.

Steps for performing a conversion

MACQUARIE University
SYDNEY·AUSTRALIA

- Check that no conversion is currently running (ADSC bit should be 0)
- If no conversion is running, start a conversion (set ADSC bit to 1)
- Wait for sample to complete (ADSC == 0)
- Read ADC to get all 16 bits (or just ADCH if you only want MSBs)

MACQUARIE University
SYDNEY·AUSTRALIA

Minor Project Q&A