

ADC



RUDRA PRATAP SUMAN

ADC: Analog to Digital Converter



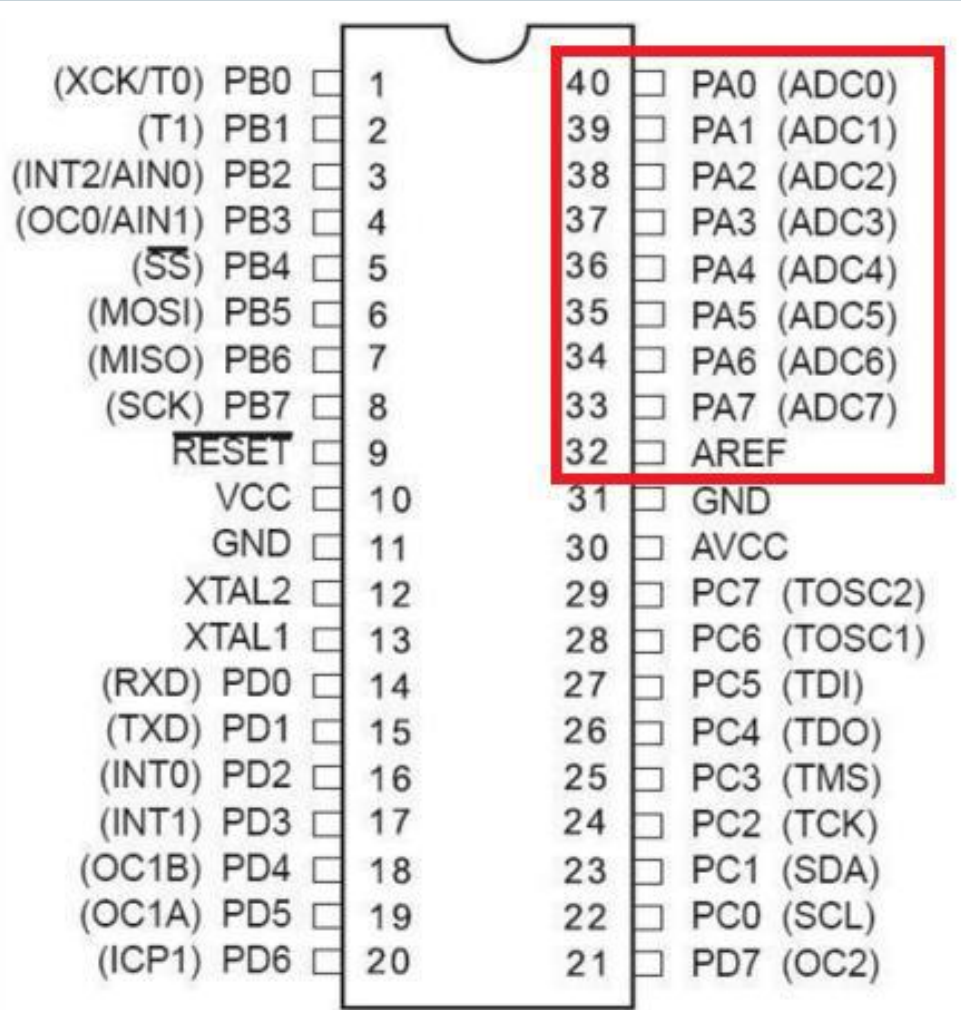
- What is analog ?
 - It is continuous range of voltage values (not just 0 or 5V)

ADC: Analog to Digital Converter

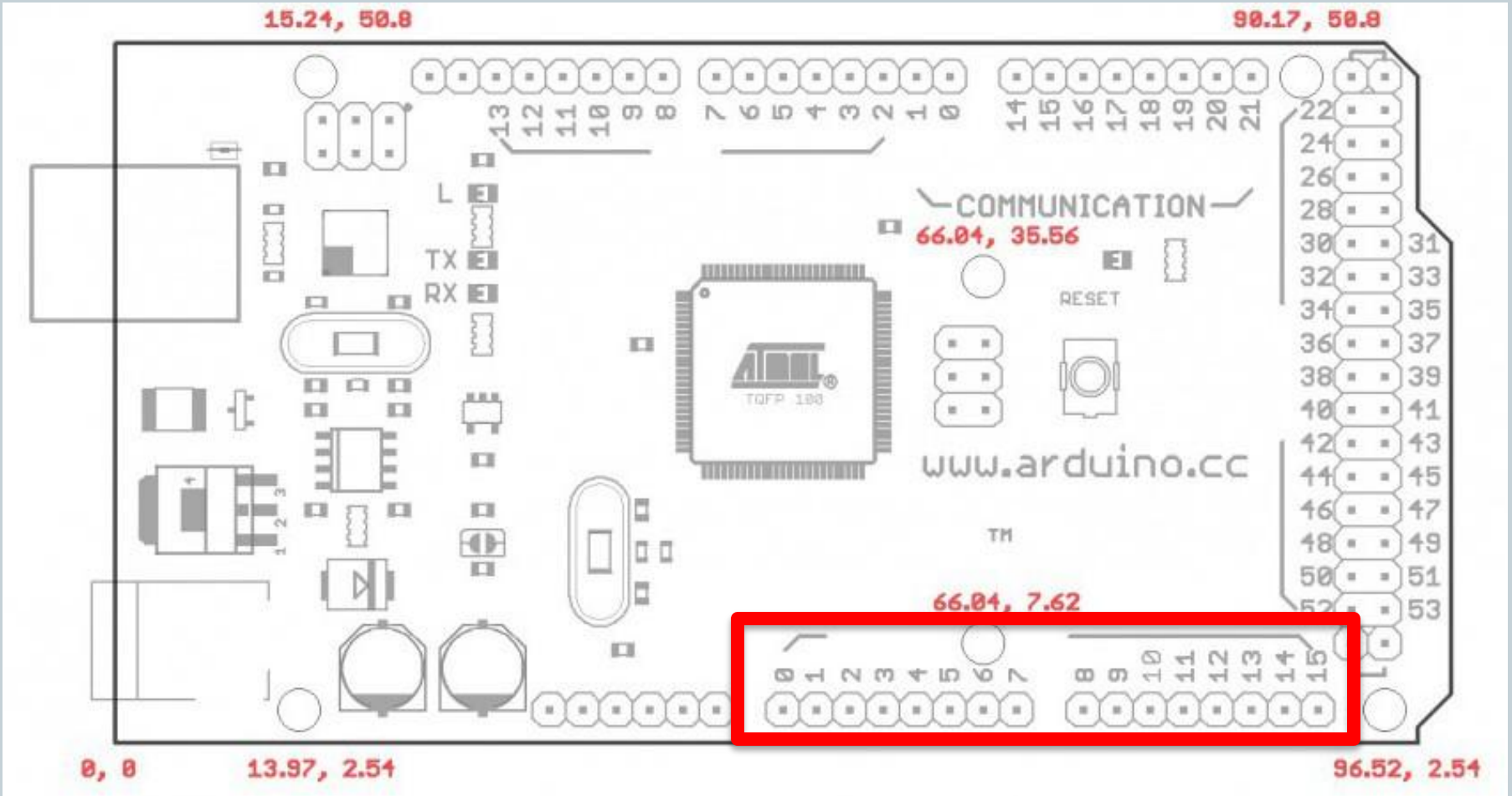


- What is analog ?
 - It is continuous range of voltage values (not just 0 or 5V)
- Why convert to digital ?
 - Because our microcontroller only understands digital.

ADC in Atmega16



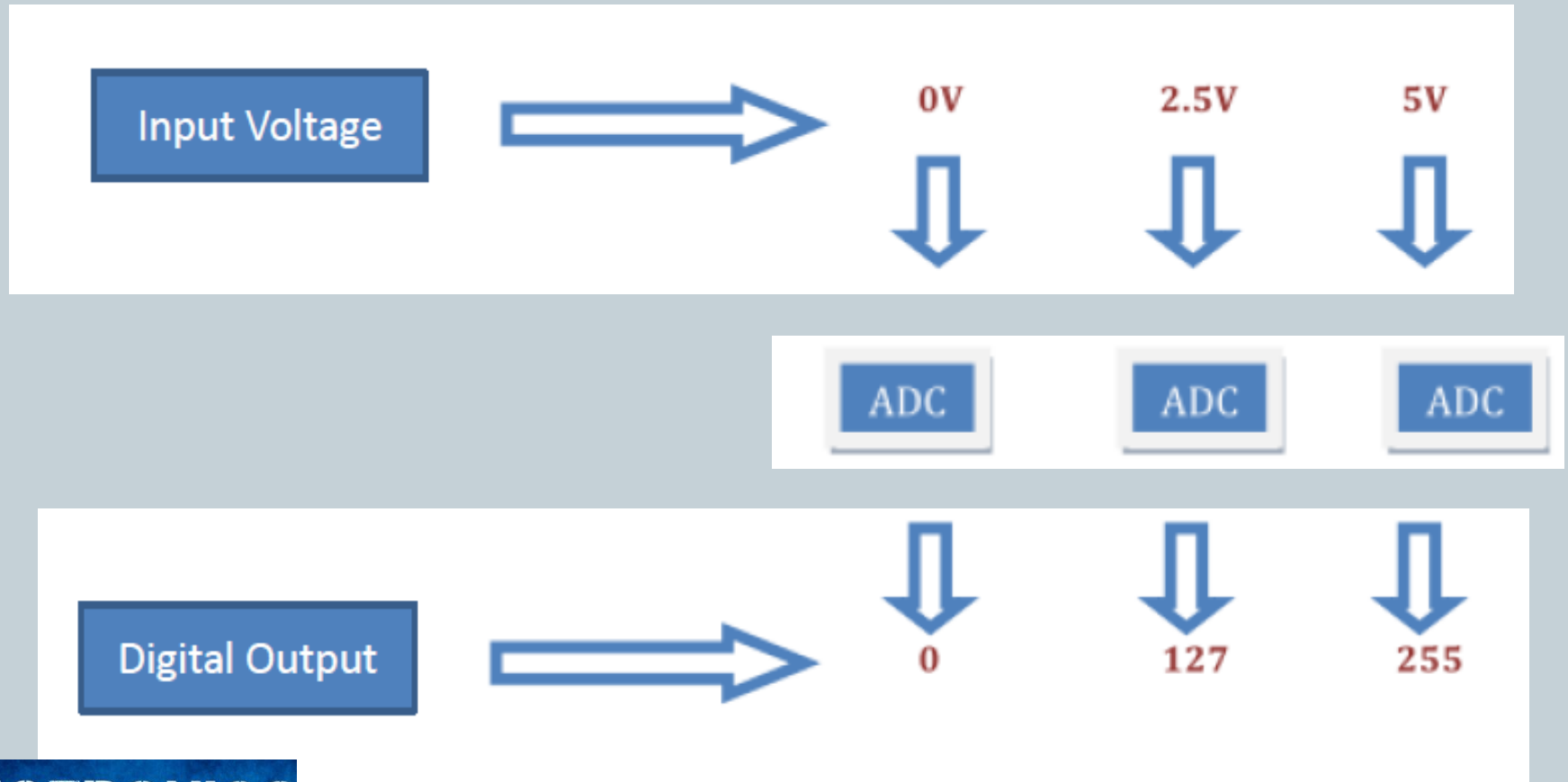
ADC in Arduino



Converting Analog Value to Digital Input



Converting Analog Value to Digital Input

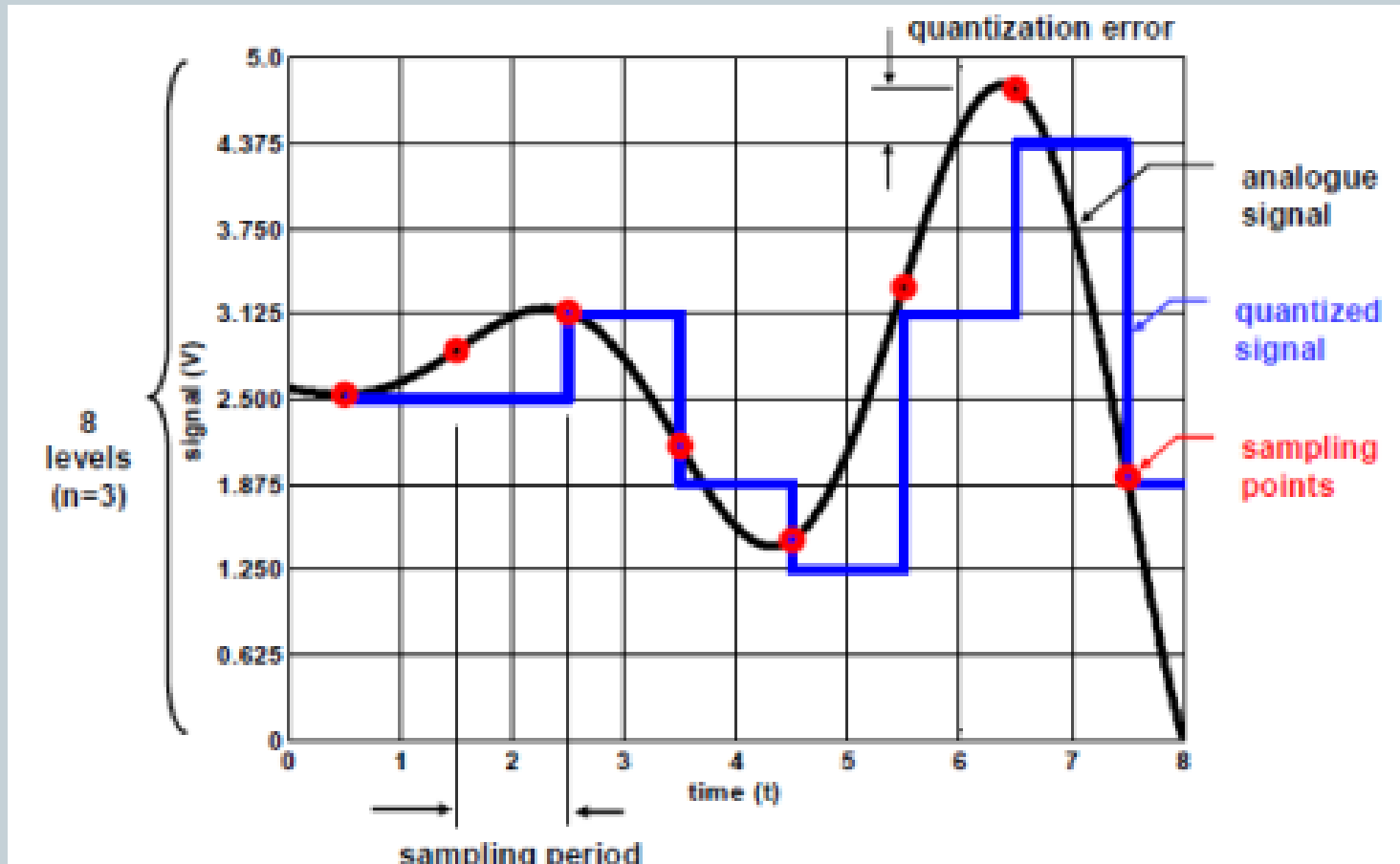


ARef

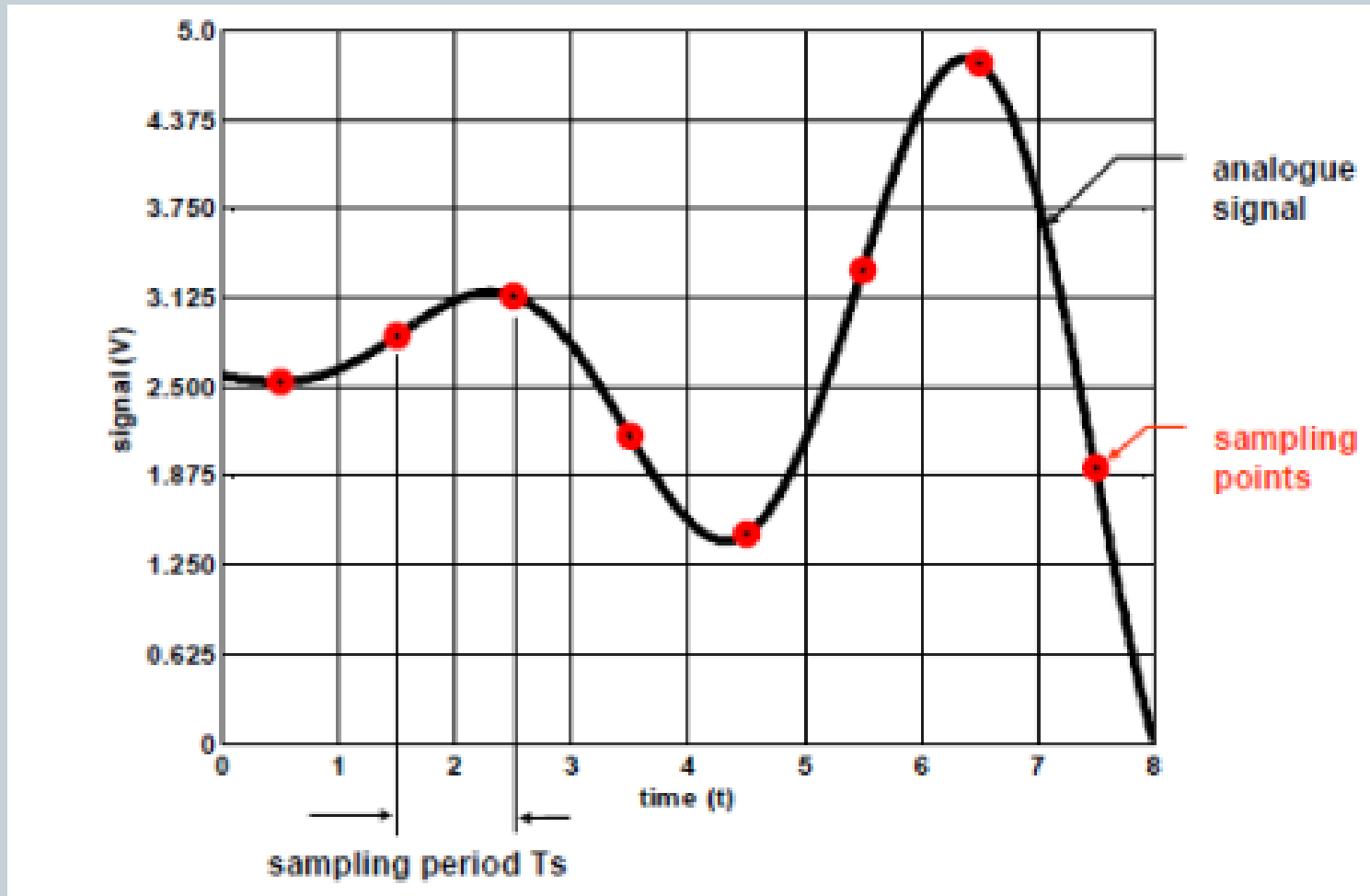


- Reference voltage corresponding to $1023 / 255$

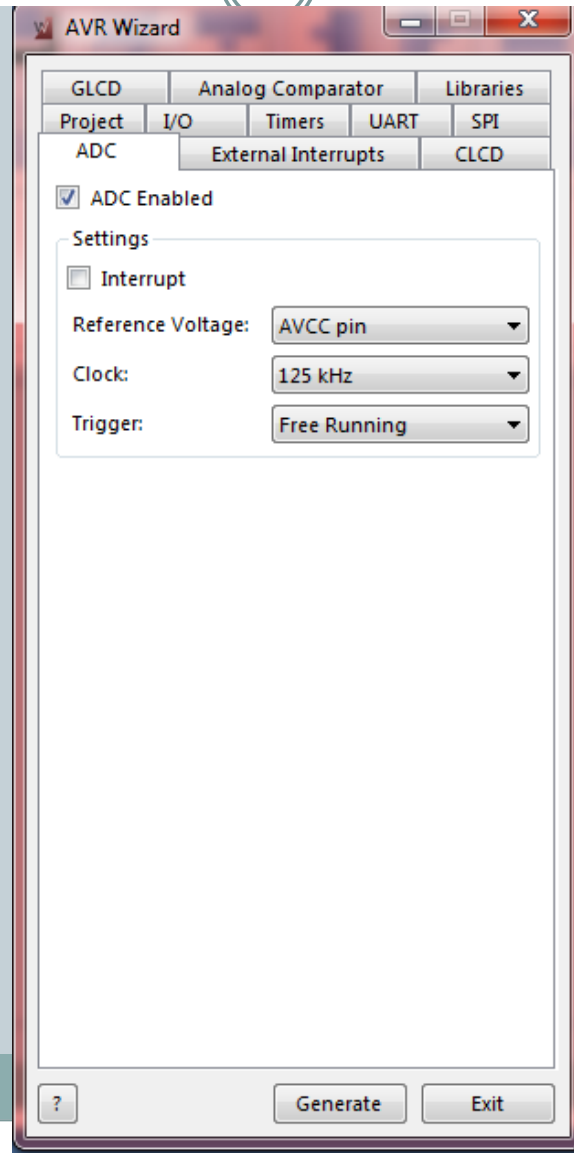
Quantizing the Signal



Sampling Rate



How To Code...



Sample Code For Atmega On AvrStudio



- It is fairly easy to obtain an ADC value. All you need is this function:
- `ADCRead(x);`
- Where 'x' is the pin number at which you are receiving the analogue output. This function returns an integer value. A sample implementation would be like:
- `a = ADCRead(5);`

Sample Code For Atmega On CVAvr



- It is fairly easy to obtain an ADC value. All you need is this function:
- `read_adc(x);`
- Where 'x' is the pin number at which you are receiving the analogue output. This function returns an integer value. A sample implementation would be like:
- `a = read_adc(5);`



- The Arduino board contains a 6 channel (8 channels on the Mini and Nano, 16 on the Mega)
- 10-bit analog to digital converter
- This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023
- It takes about 100 microseconds (0.0001 s) to read an analog input
- 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit



- The input range and resolution can be changed using [analogReference\(\)](#)
 - DEFAULT: the default analog reference of 5 volts (on 5V Arduino boards) or 3.3 volts (on 3.3V Arduino boards)
 - INTERNAL: an built-in reference, equal to 1.1 volts on the ATmega168 or ATmega328 and 2.56 volts on the ATmega8 (*not available on the Arduino Mega*)
 - INTERNAL1V1: a built-in 1.1V reference (*Arduino Mega only*)
 - INTERNAL2V56: a built-in 2.56V reference (*Arduino Mega only*)
 - EXTERNAL: the voltage applied to the AREF pin (**0 to 5V only**) is used as the reference.



Warning

- **Don't use anything less than 0V or more than 5V for external reference voltage on the AREF pin! If you're using an external reference on the AREF pin, you must set the analog reference to EXTERNAL before calling analogRead().** Otherwise, you will short together the active reference voltage (internally generated) and the AREF pin, possibly damaging the microcontroller on your Arduino board.



Solution

- Alternatively, you can connect the external reference voltage to the AREF pin through a 5K resistor, allowing you to switch between external and internal reference voltages.
- Note that the resistor will alter the voltage that gets used as the reference because there is an internal 32K resistor on the AREF pin. The two act as a voltage divider, so, for example, 2.5V applied through the resistor will yield $2.5 * 32 / (32 + 5) = \sim 2.2V$ at the AREF pin.



Thank You
Question??