VIA UNIVERSITY COLLEGE

# Interfacing to the Analogue World

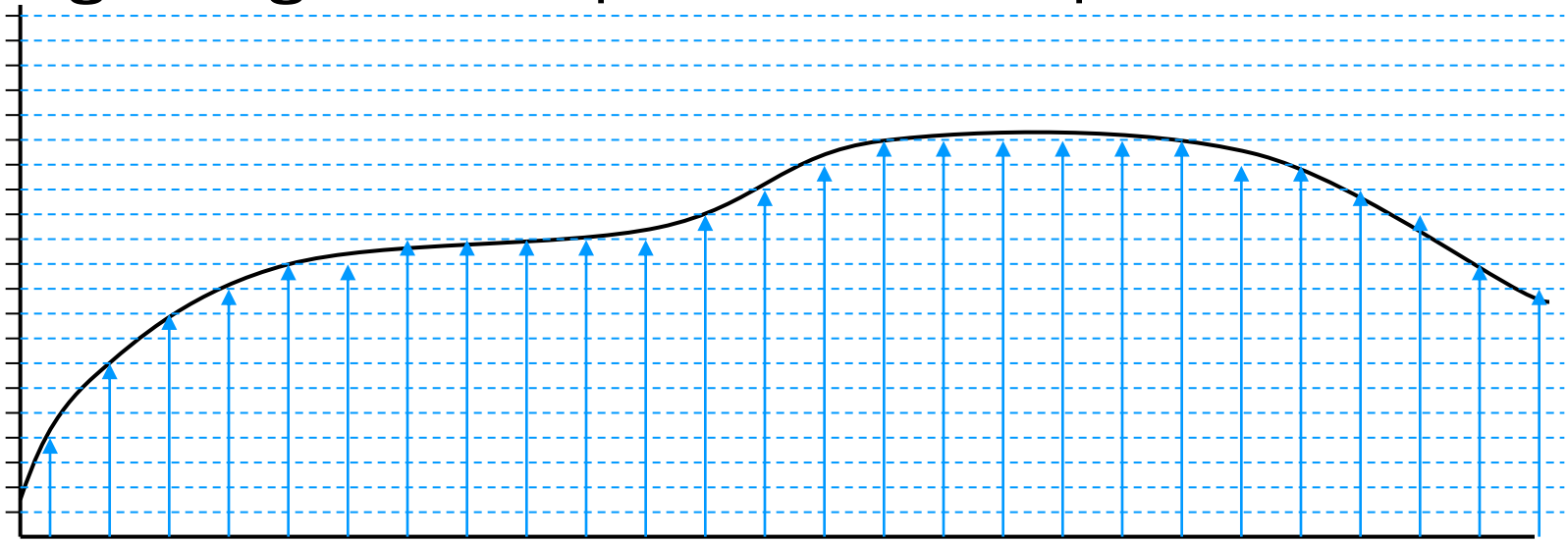
# (HWP I1)

# Analogue interfacing

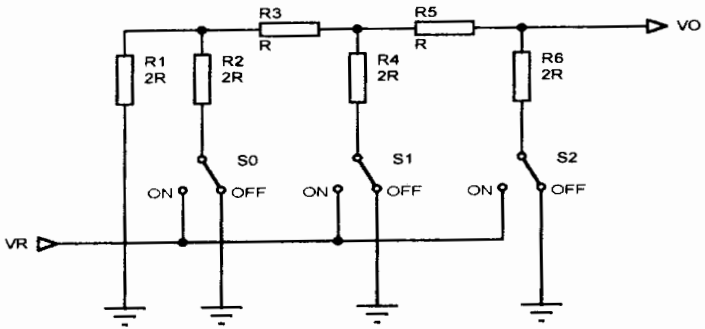
- D/A-converters
- A/D-converters
- Pulse Width Modulation (PWM)
- Etc.

# Analogue versus Digital

- Analogue signals are continuously
- Digital signals are quantified in steps



# D/A-Converters



# D/A-Converters

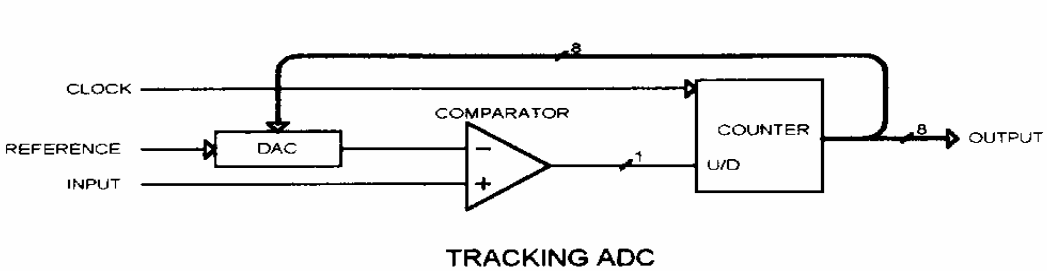
| *S2* | *S1* | *S0* | *VO* | |
|---|---|---|---|---|
| OFF | OFF | OFF | 0 | |
| OFF | OFF | ON | .125 × VR | (1/8 × VR) |
| OFF | ON | OFF | .25 × VR | (2/8 × VR) |
| OFF | ON | ON | .375 × VR | (3/8 × VR) |
| ON | OFF | OFF | .5 × VR | (4/8 × VR) |
| ON | OFF | ON | .625 × VR | (5/8 × VR) |
| ON | ON | OFF | .75 × VR | (6/8 × VR) |
| ON | ON | ON | .875 × VR | (7/8 × VR) |

# A/D-Converters

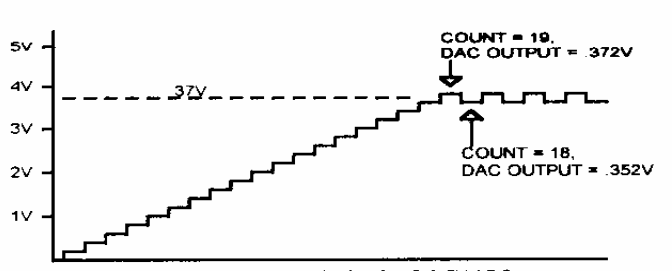
- Tracking A/D-Converters
- Flash A/D-Converters
- Successive Approximation A/D-Converters

- Sample/Hold

# Tracking A/D-Converter



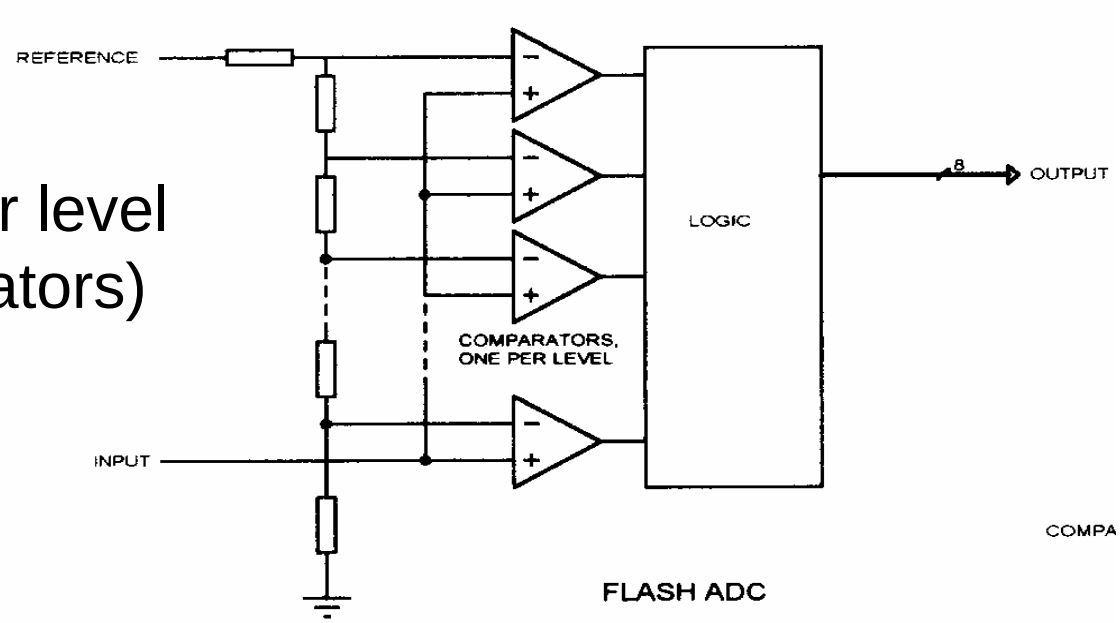
TRACKING ADC



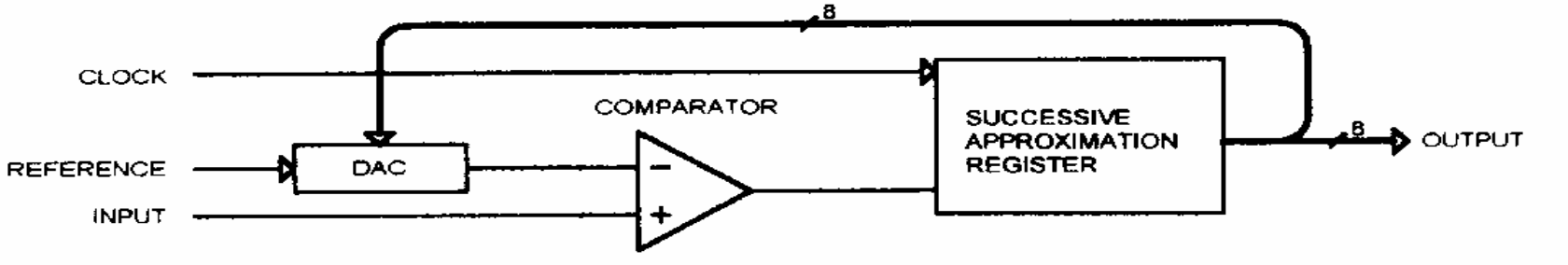

- Slow if signal are fluctuating

# Flash A/D-Converter

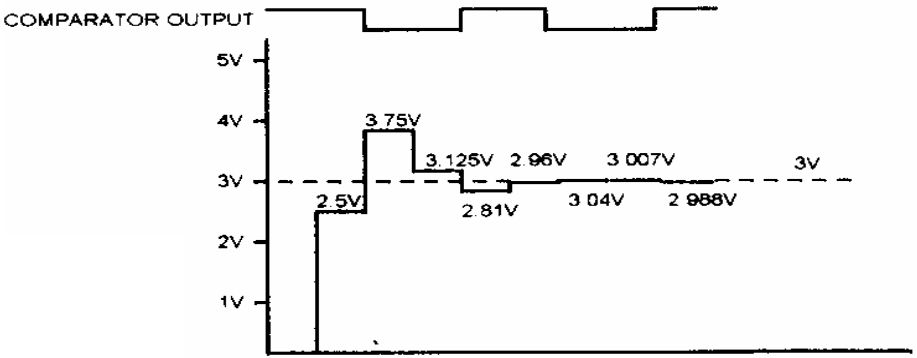
- Very Fast
- Expensive
  - One comparator per level (4bit => 16 comparators)



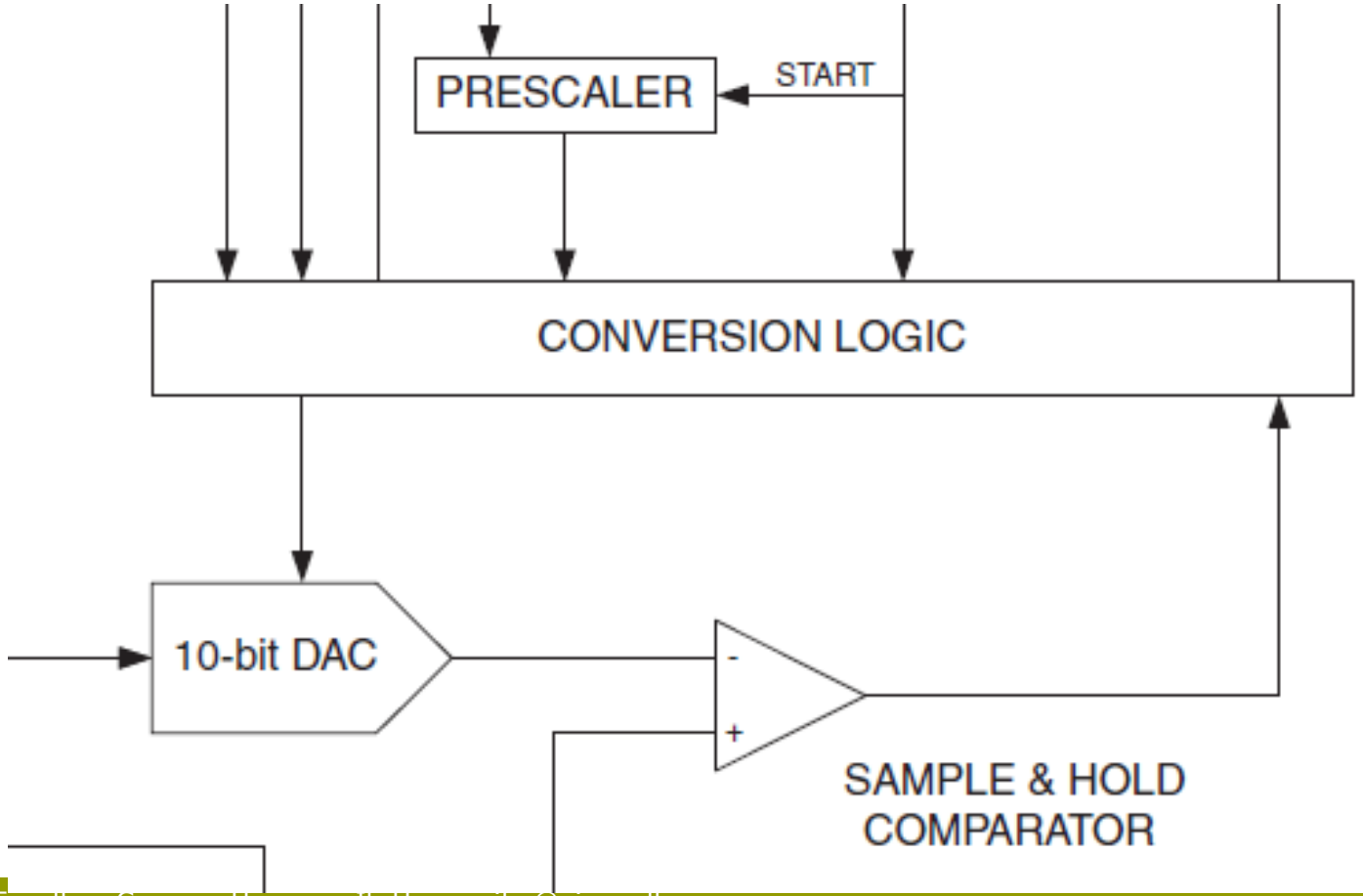
# Successive Approximation A/D-Converter




- Most used type
- Medium fast
- Cheap

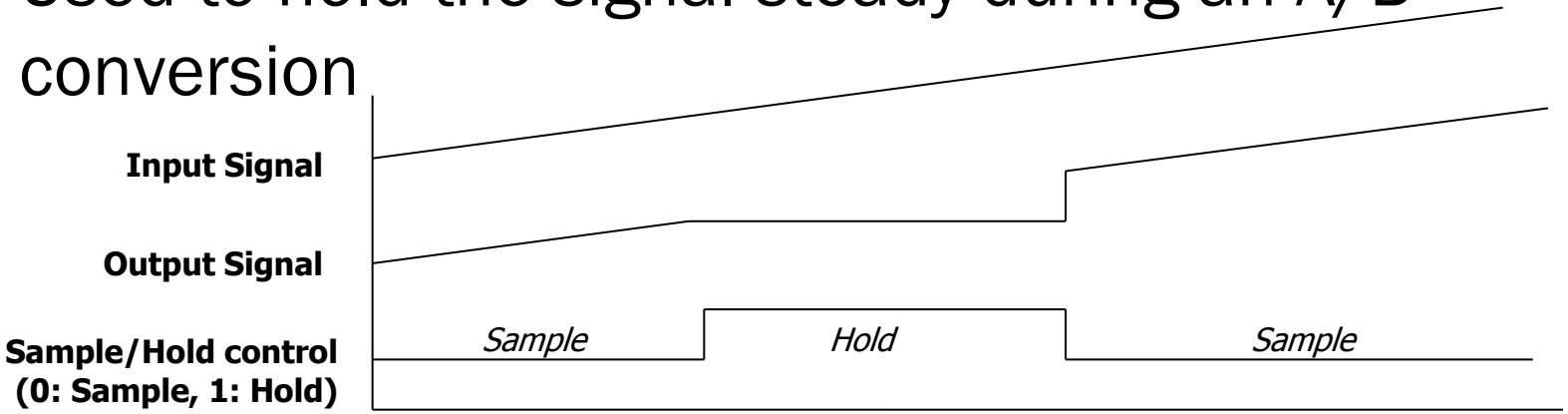


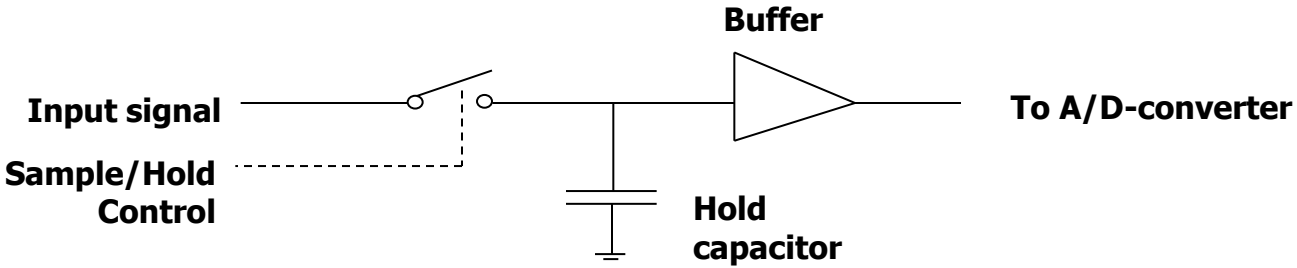
# In the ATMEGA1280



# Sample/Hold

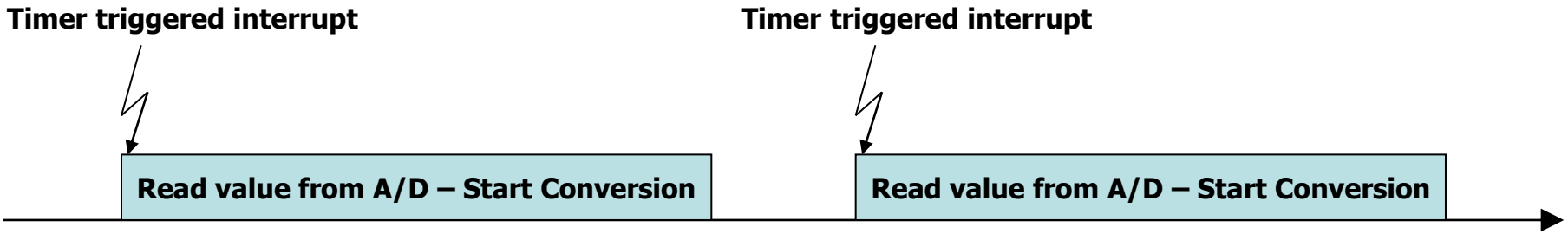
- Stabilize analogue signals for short periods
- Used to hold the signal steady during an A/D-conversion



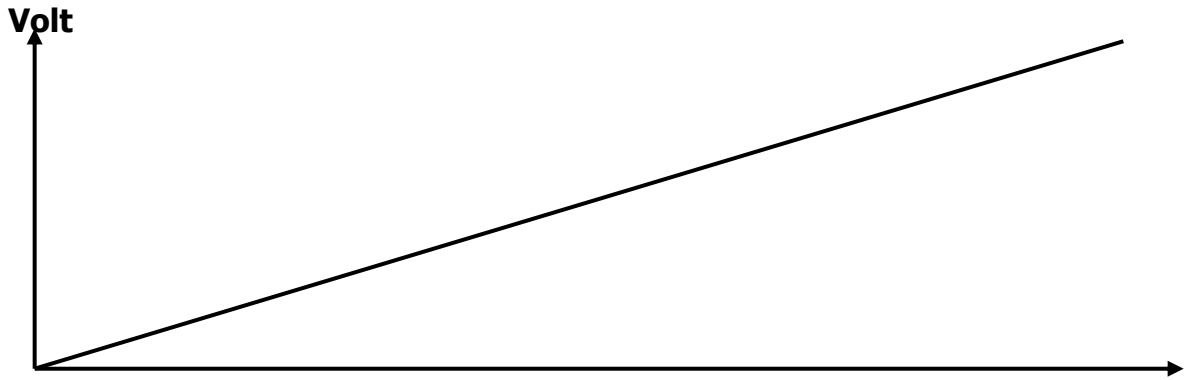
# The software for A/D-conversion

- We want to sample with a constant period between samples



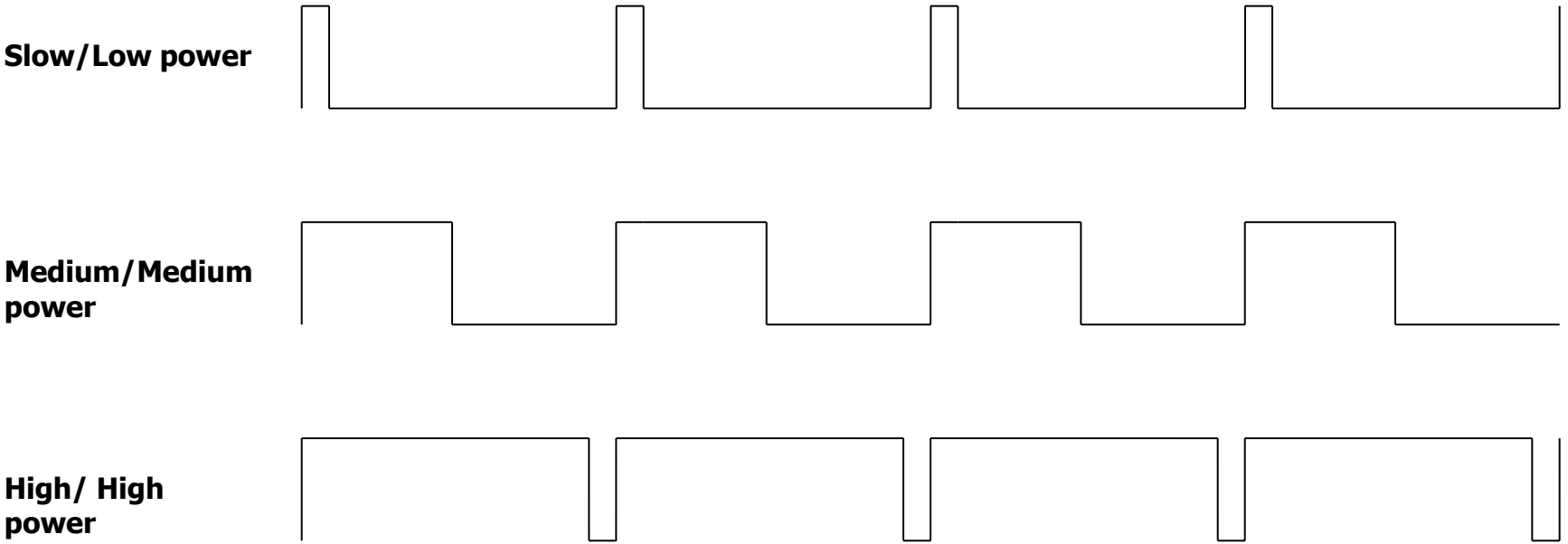
# Pulse Coded Modulation (PCM/PWM)

Analogue control of motor speed, heater etc.



# Pulse Coded Modulation (PCM)

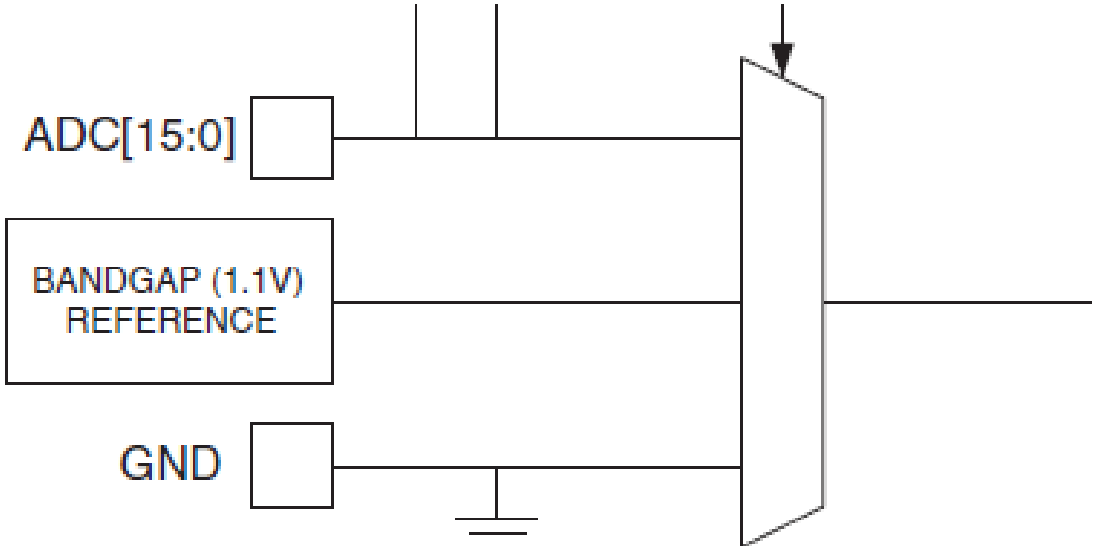
PCM control of motor speed, heater etc.



# AVR

- Read about the A/D converter in the documentation for the ATMEGA1280 MCU
- Exercise:
  - How many modes can the ADC be setup to
  - If some music (20-20khz) must be sampled - how would you setup the ADC?
  - Setup the ADC to measure a voltage between 0 and 2.5V

# ATMEGA1280 ADC Mux

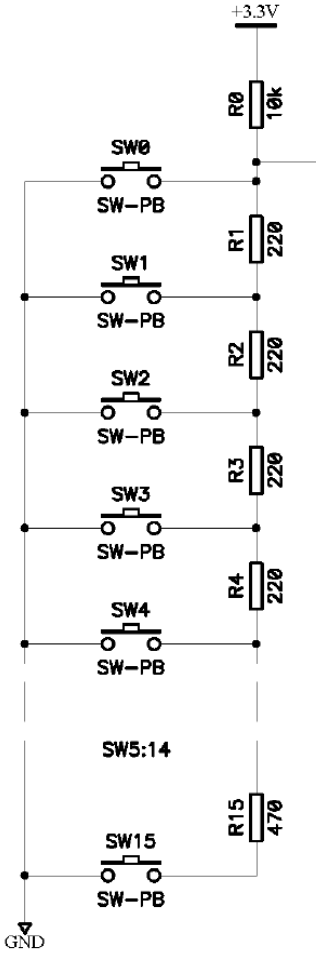


# Discussion

- If you have 16 keys/switches that you must interface to, can this be done with a single ADC input?

- If it can how?

# Solution




The input voltage associated with each key n is given by:

$$Vin = (Vdd - Vss) \times \Sigma(R1..Rn) / (R0 + \Sigma(R1..Rn))$$

If more than one key is pressed at the same time, the key detected is the closest key to the ADC input in the chain. This means that the key recognition is managed by priority.