

PV198 - Analog to Digital Conversion

One-chip Controllers

Dávid Danaj, Marek Vrbka

Faculty of Informatics, Masaryk University

Content

Overview

Demo

Theory

Successive Approximation

Joystick

Application

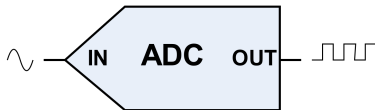
Homework

Intro

- Switch the branch!

What is ADC

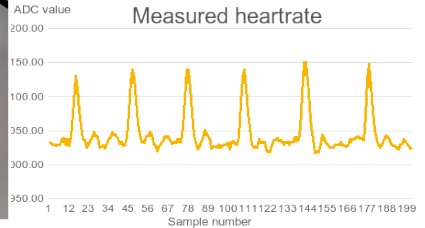
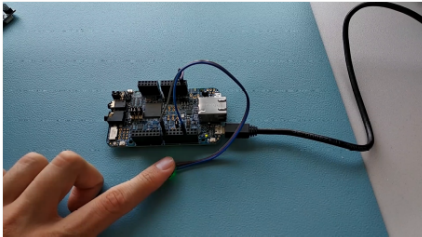
- **ADC** - Analog to **D**igital **C**onverrter
- Convert analog signal into digital values



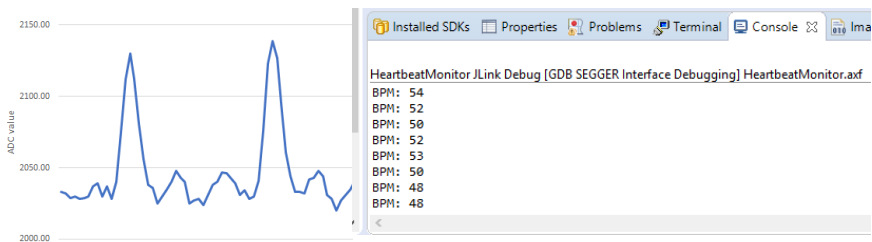
What is it used for

- Everything that needs to transfer analog values into digital values
- Audio
- Sensors
- Video

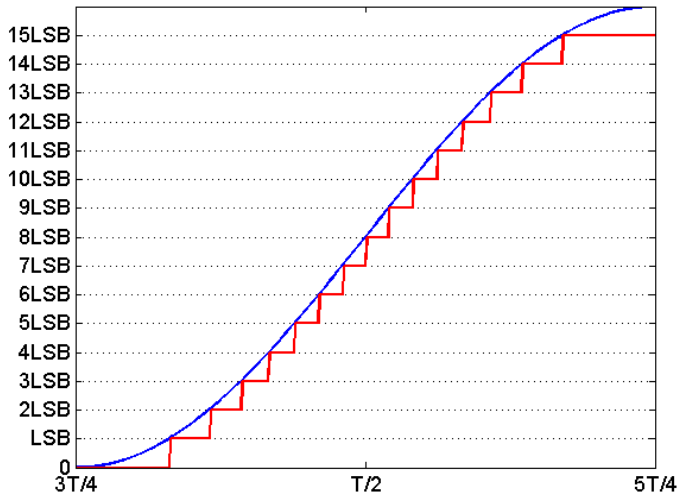
Demo - Heartbeat sensor



Demo - Heartbeat sensor

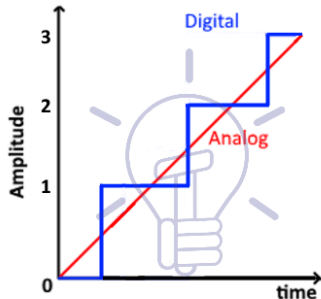


How does it work

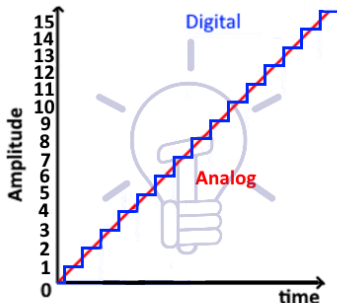


Sampling resolution

- Smallest incremental voltage that can be recognized and thus causes a change in the digital output
- Expressed as the number of bits output by the ADC



2-bit Resolution



4-bit Resolution

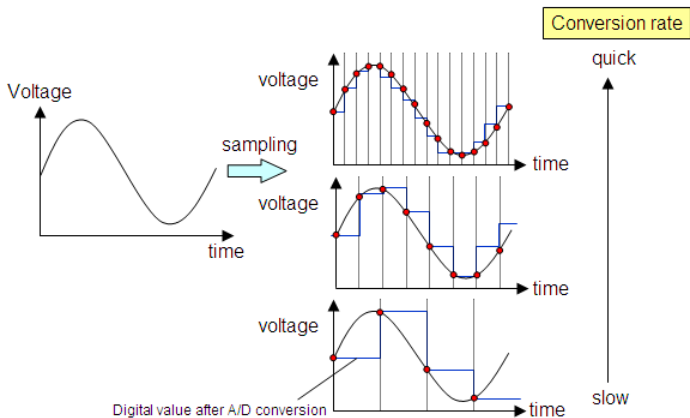
<https://www.electricaltechnology.org/2019/02/analog-to-digital-converter-adc.html>

Quantization Error

- Difference between analog value and rounded digital value (rounding error)
 - Change in analog value smaller than the step of the digital value is ignored
 - The quantization error is 0.5 least significant bit for the ADC

Sampling Rate

- Nyquist-Shannon theorem
- ADC sampling frequency must be at least twice the analog signal frequency



Dynamic Range

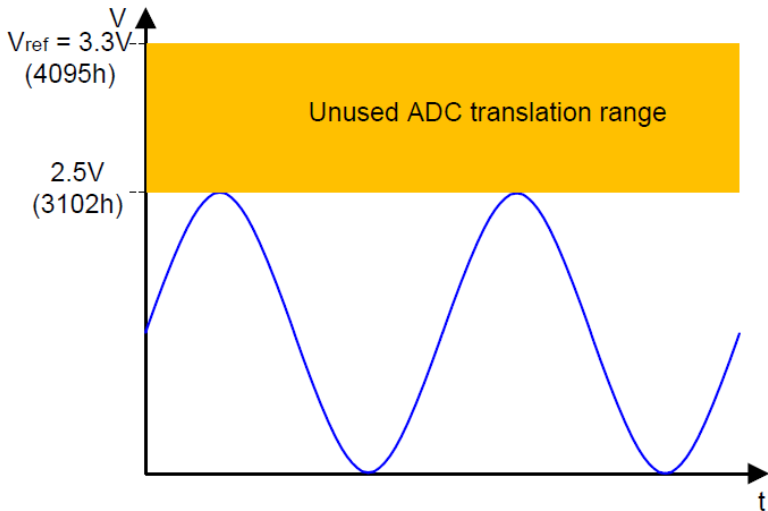
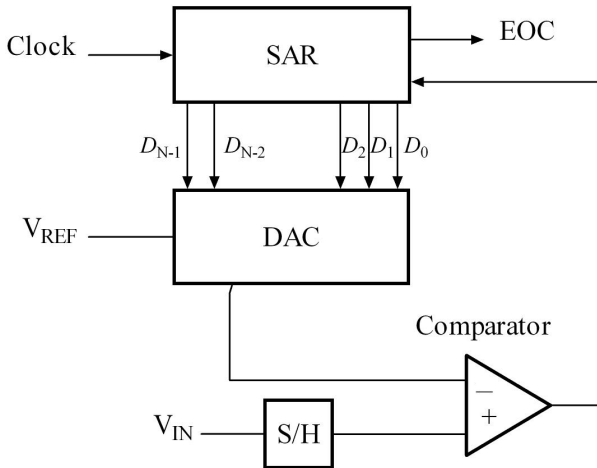


Figure 9. ADC dynamic range

Linear Successive Approximation Algorithm



Linear Successive Approximation Algorithm

- $V_{in} = 2.1V$
- $V_{ref} = 3.3V$
- 3-bit resolution

Step	Digital code	DAC output	Output Comparator	Digital output
1	100	1.65V	1	1 00
2	110	2.475V	0	10 0
3	101	2.0625V	1	101

Linear Successive Approximation Algorithm

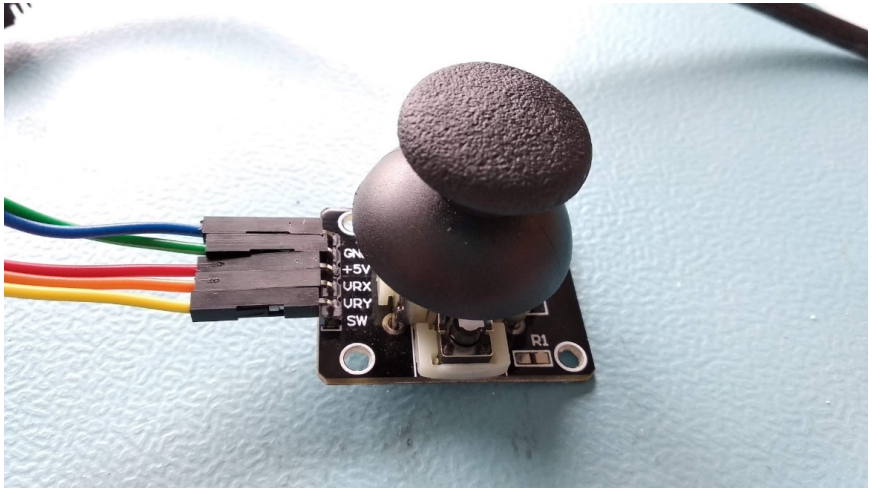
- $V_{in} = 1.16V$
- $V_{ref} = 3.3V$
- 5-bit resolution

Step	Digital code	DAC output	Output Comparator	Digital output
1				
2				
3				
4				
5				

FRDM-K66F ADC

- Linear successive approximation algorithm with up to 16-bit resolution
- Single or continuous conversion
- Can work in low-power modes
- HW trigger to start conversion
- HW average function
- Conversion complete interrupt
- DMA support

Joystick



Joystick

■ DO NOT USE 5V

3.6.1.1 16-bit ADC operating conditions

Table 31. 16-bit ADC operating conditions

Symbol	Description	Conditions	Min.	Typ. ¹	Max.	Unit	Notes
V_{DDA}	Supply voltage	Absolute	1.71	—	3.6	V	—
ΔV_{DDA}	Supply voltage	Delta to V_{DD} ($V_{DD} - V_{DDA}$)	-100	0	+100	mV	2
ΔV_{SSA}	Ground voltage	Delta to V_{SS} ($V_{SS} - V_{SSA}$)	-100	0	+100	mV	2
V_{REFH}	ADC reference voltage high		1.13	V_{DDA}	V_{DDA}	V	
V_{REFL}	ADC reference voltage low		V_{SSA}	V_{SSA}	V_{SSA}	V	
V_{ADIN}	Input voltage	<ul style="list-style-type: none"> 16-bit differential mode All other modes 	V_{REFL}	—	$\frac{31}{32} \cdot V_{REFH}$ V_{REFH}	V	—

Joystick - Connection (Easy)

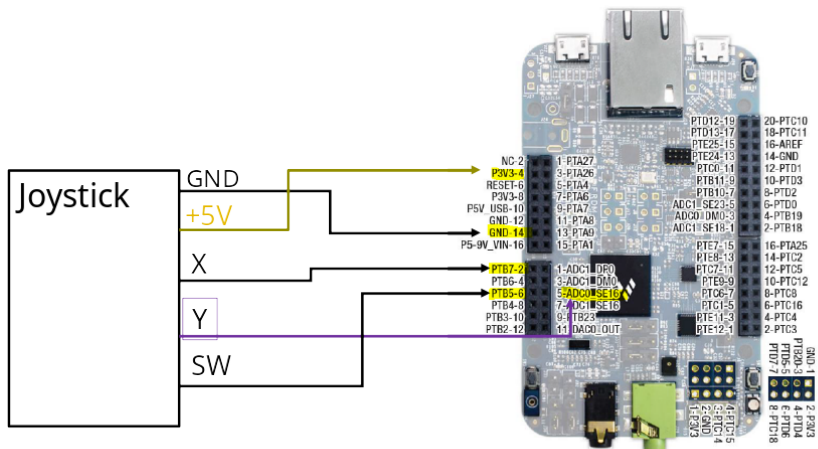


Figure 26. FRDM-K66F I/O header pinout

Application

Tasks

- Create an application that reads position of the joystick (both axes) using ADC0 and ADC1
- Print position to terminal
- Write periodic check into while loop if measurement is finished
- Use PIT to initialize read only once a second
- Either
 - Write ADC read only using interrupts
 - React to joystick button press

Steps

- Initialize (MCUXpresso Configuration Tools might help here)
 - Pins
 - Clocks
 - Peripherals (configure ADC0 and ADC1)
- Write application code

Homework - Use joystick to control LEDs

- Only use ADC1 peripheral with multiple channels
- Warning: ADC has special handling of interrupt flags
- Use the same connection as shown in the difficult slide
- PTB7 controls green LED, PTB6 controls red LED

MUNI

FACULTY

OF INFORMATICS