TTK4155

Industrial and Embedded Computer Systems Design

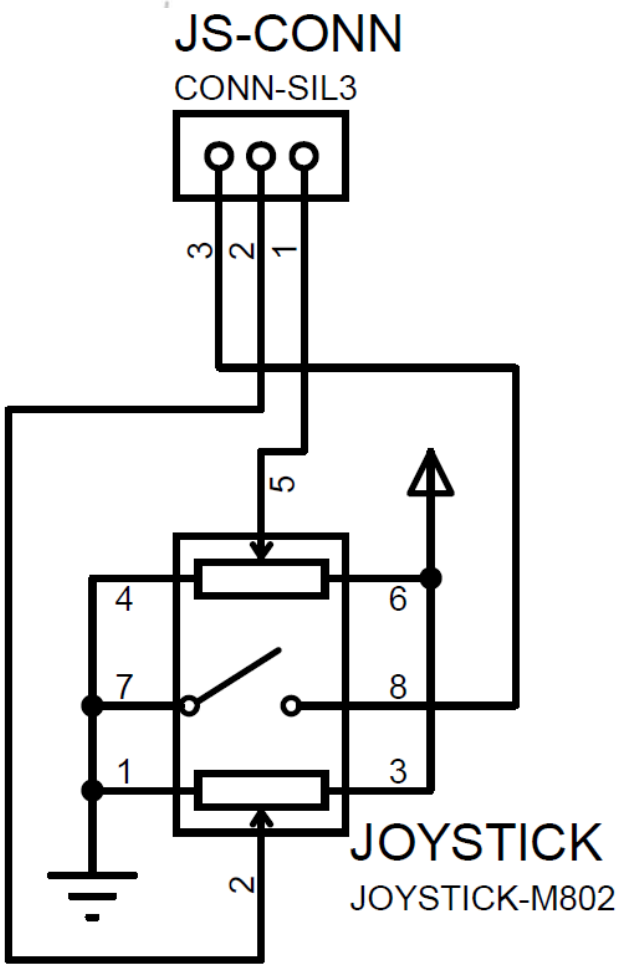
NTNU – Trondheim
Norwegian University of
Science and Technology

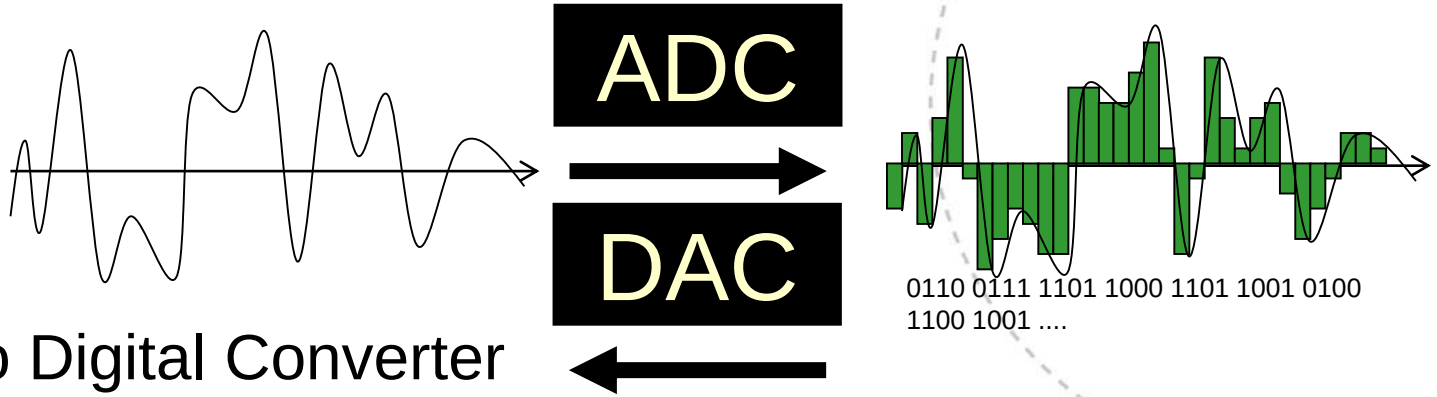
Lab lecture 3

- A/D Conversion
- Joystick input
- Sliders and PWM

Joystick

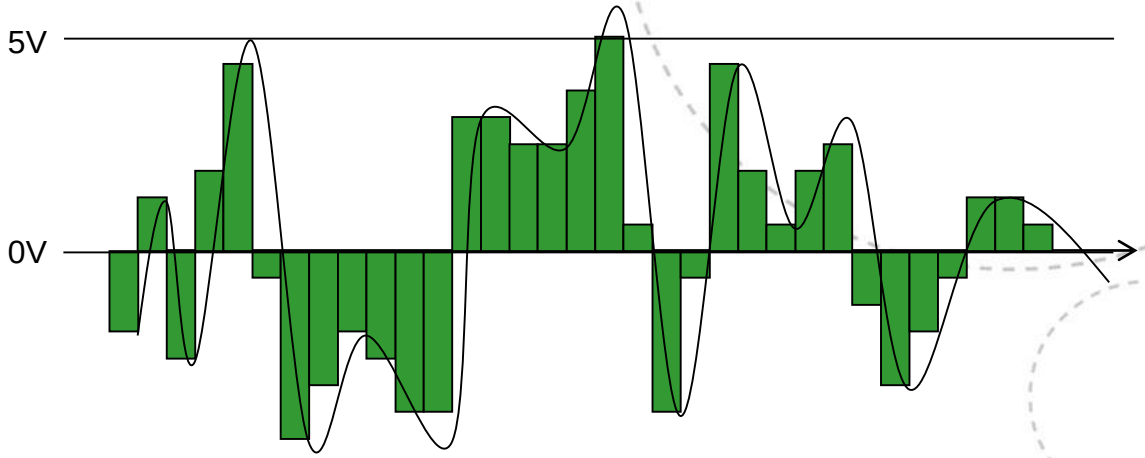
- Two axis (X, Y)
  - Variable resistance
- One button
  - Short circuiting
- How to read position and buttons?
  - Button => PINx & 0x01 maybe





- Analog to Digital Converter
- **Analog** voltage → **digital** value
  - 0 V – 5 V
  - 0000 0000 – 1111 1111 (0 – 255)
- Continuous signals → discrete, quantified signals
- Reverse: DAC
- MCUs may have an ADC built-in
- Examples
  - Record sound
  - Read analog sensors
  - Joystick (!)
  - PWM with LP filter etc.

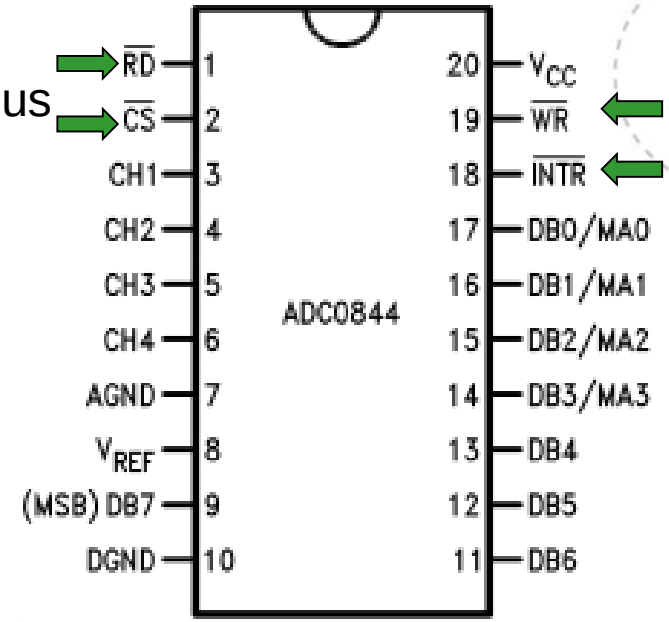
ADC parameters



- Number of bits → number of levels
  - 8 bit → 2^8 = 256 levels
- Voltage span
  - 0 – 5V
- Together, this gives the resolution
  - 5V / 256 = 20 mV
- Quantization error

ADC in this project

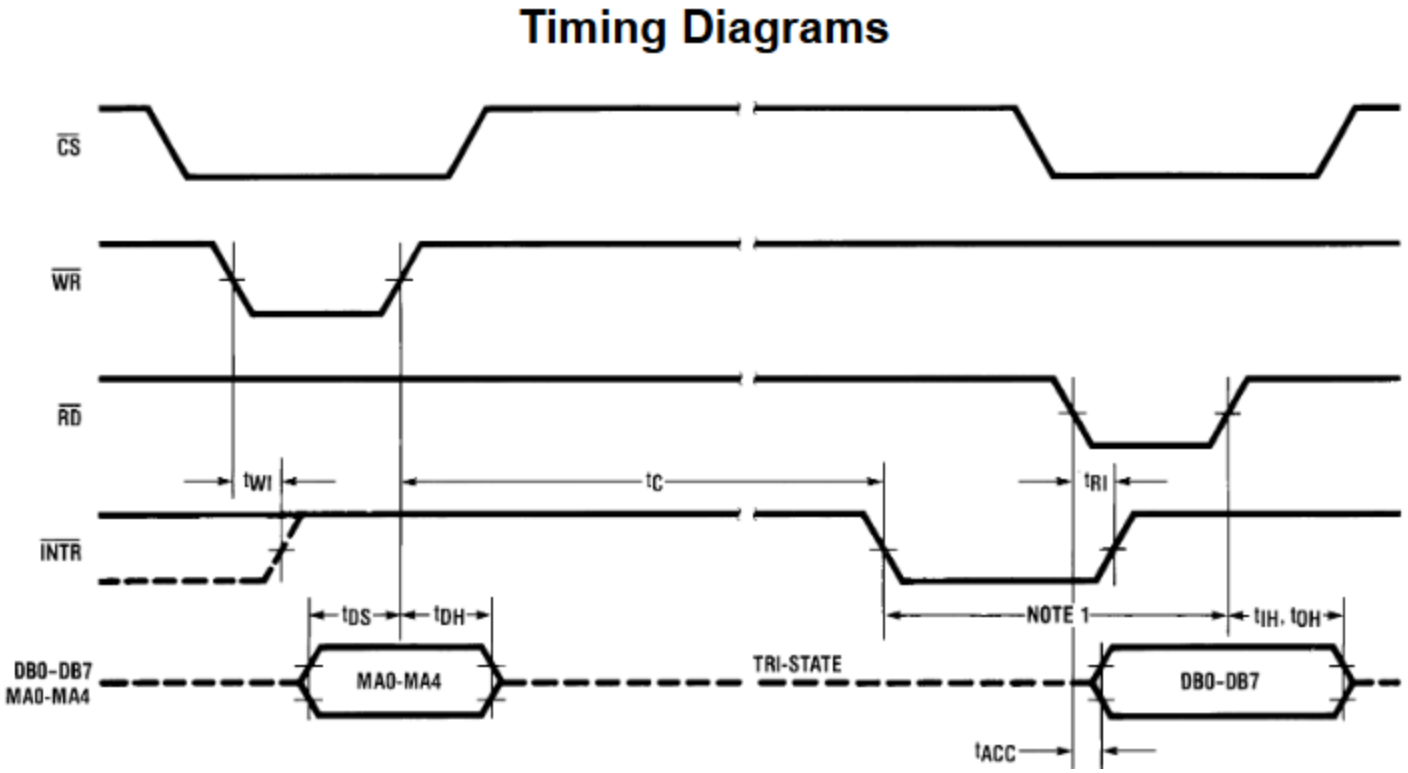
- ADC0844CNN from National Semiconductor
  - 4 channels, SAR 8-bit ADC
  - Read analog value: 40 µs
  - 8 bit parallel data and mux address latch bus
  - CS
  - Single ended / differential
  - Interrupt output



Reading from ADC

- Access the ADC via the memory space.
- Volatile – ”others might be changing our data”
- Steps to read ADC:
  1. Configure the channel/mode.
     - Single-ended or differential?
     - Select mode by writing to the data bus. Set /WR low.
  2. Read the ADC results after /INTR goes low. Set /RD low to start reading.
- Interrupt based approach is recommended.

ADC timing diagram



/WR and /RD => outputs from AVR
/INTR => input to AVR
/CS => output from GAL

Drivers

- Make separate ADC and joystick drivers.
- ADC interface:
  - byte ADC_read(byte channel);
- Joystick interface:
  - void JOY_init();
  - void JOY_calibrate();
  - bool JOY_button(int button);
  - JOY_position_t JOY_getPosition(); → eg. X: 83%, Y: -21%
  - JOY_direction _t JOY_getDirection(); → LEFT

Tips

- Single ended ADC since common ground.
- Sliders => output is PWM use on-board LP filter and connect with ADC.
- structs for joystick position and direction.
- Send values via UART to display on terminal.
- Calibrate the joystick (max, min, direction…)
  - Auto calibrating in software?

Common SRAM problems

- NOISE! Decoupling, ground loops…
- Floating /CS pin.
- Wrong connection => too many wires, defected holes in breadboard.
- Use A0-A11 NOT only A0-A7. Also SRAM A11 and/or A12 should be connected to GND not left floating.
- ALE => might need LP filter.
- Check GAL logic by providing dummy addresses and using Oscilloscope or DMM.

Auto calibration

- Auto-calibration for y axis.
  - Mean_y=0.5*Range_y
    - Range_y -> variable, calibration procedure finds this.
  - Range_y=Ry_max-Ry_min
  - Assume ADC gives 25 for Ry_min & 200 for Ry_max.
  - Which means 0% = 25 & 100%=200 (ADC values)
  - In auto calibration, you can use a push button as control input and find Ry_min and Ry_max by moving joystick from one extreme to other in the initialization procedure.
- Same procedure can be followed for x axis.

Questions?

Auf wierdersehen