COMPSYS723 Embedded Systems Design

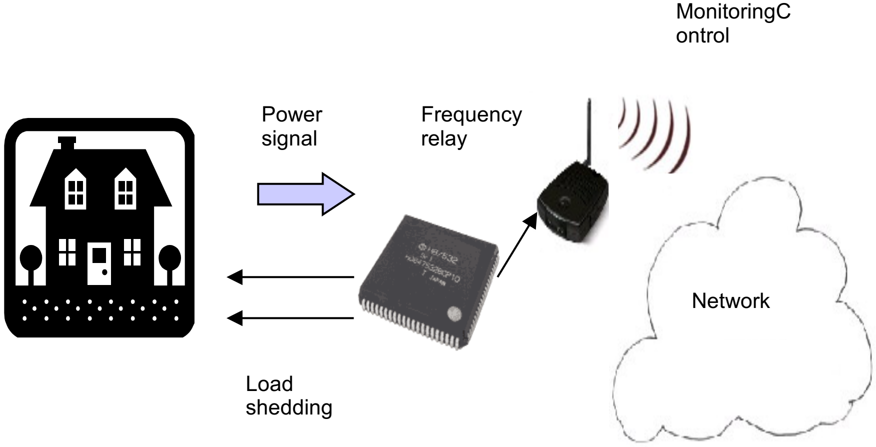
Assignment 1 – Low-Cost Frequency Relay

Zoran Salcic
Auckland University

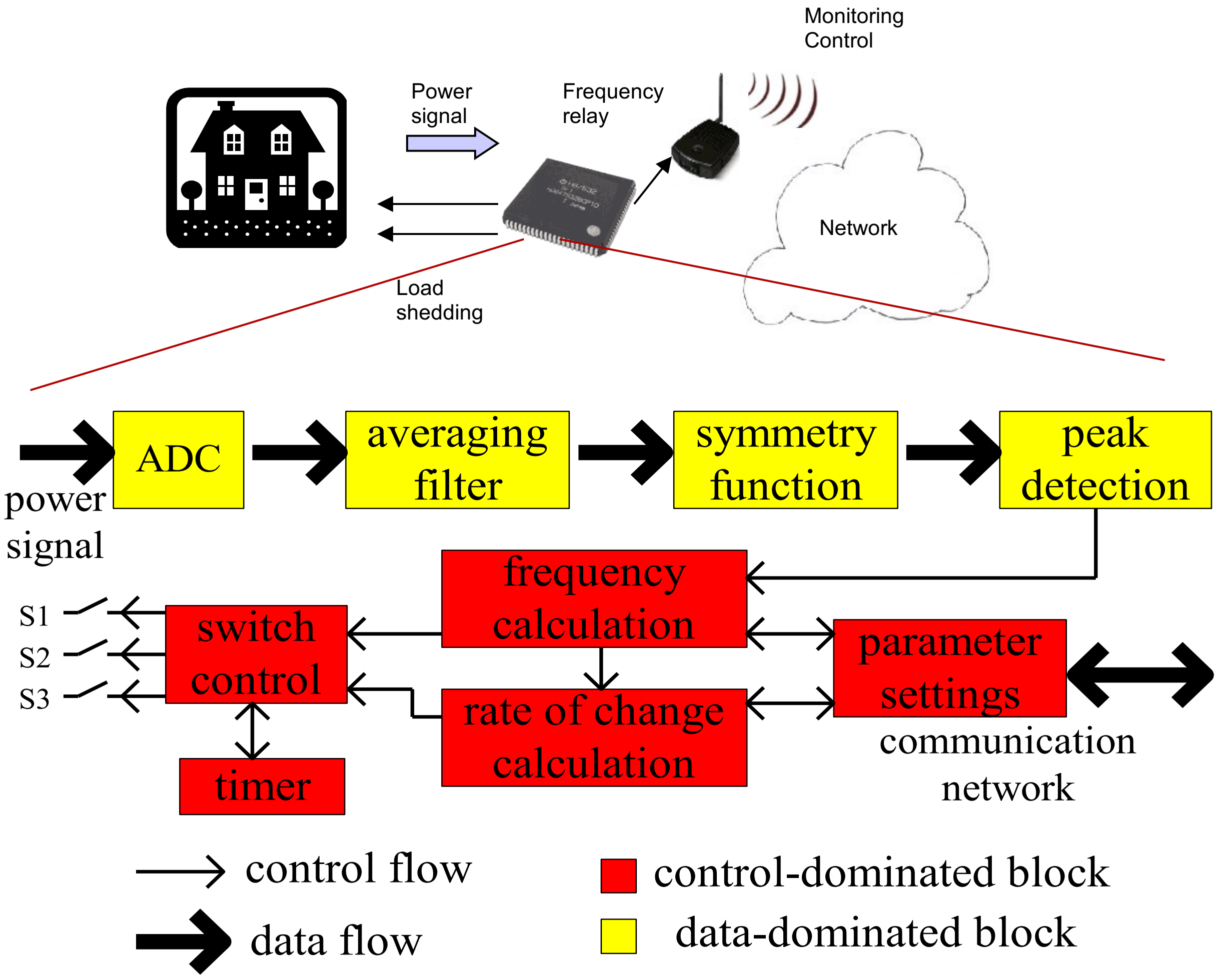
Semester 1, 2024

Internet-enabled Frequency Relay
Example of a Heterogeneous Embedded System

- Measures current frequency and its rate of change in power signal in real-time
- Sheds the loads if necessary (switches loads off and back on)
- Communicates with other supervisory and control system through landline or wireless system (higher level control)
- Operates non-stop, 24/7
- Low-cost for use at household level (under $100)

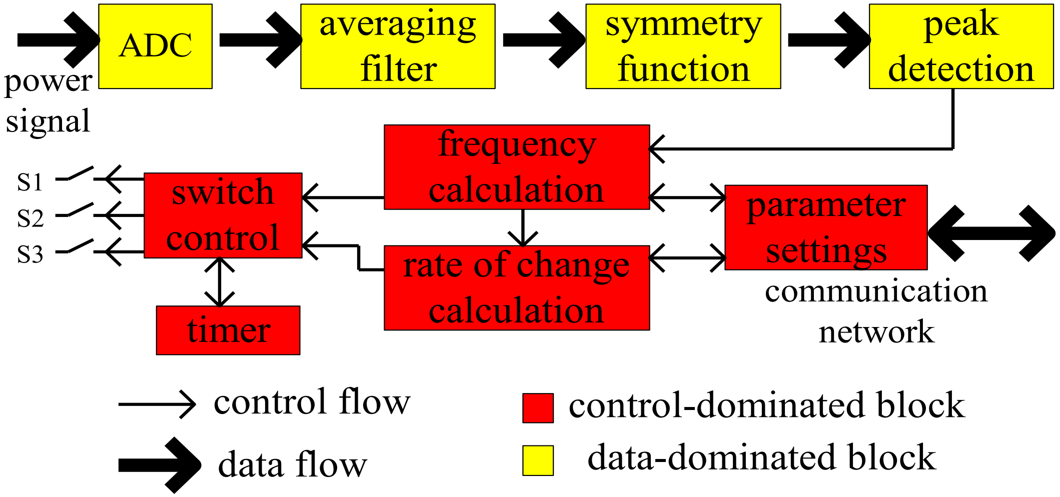


Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

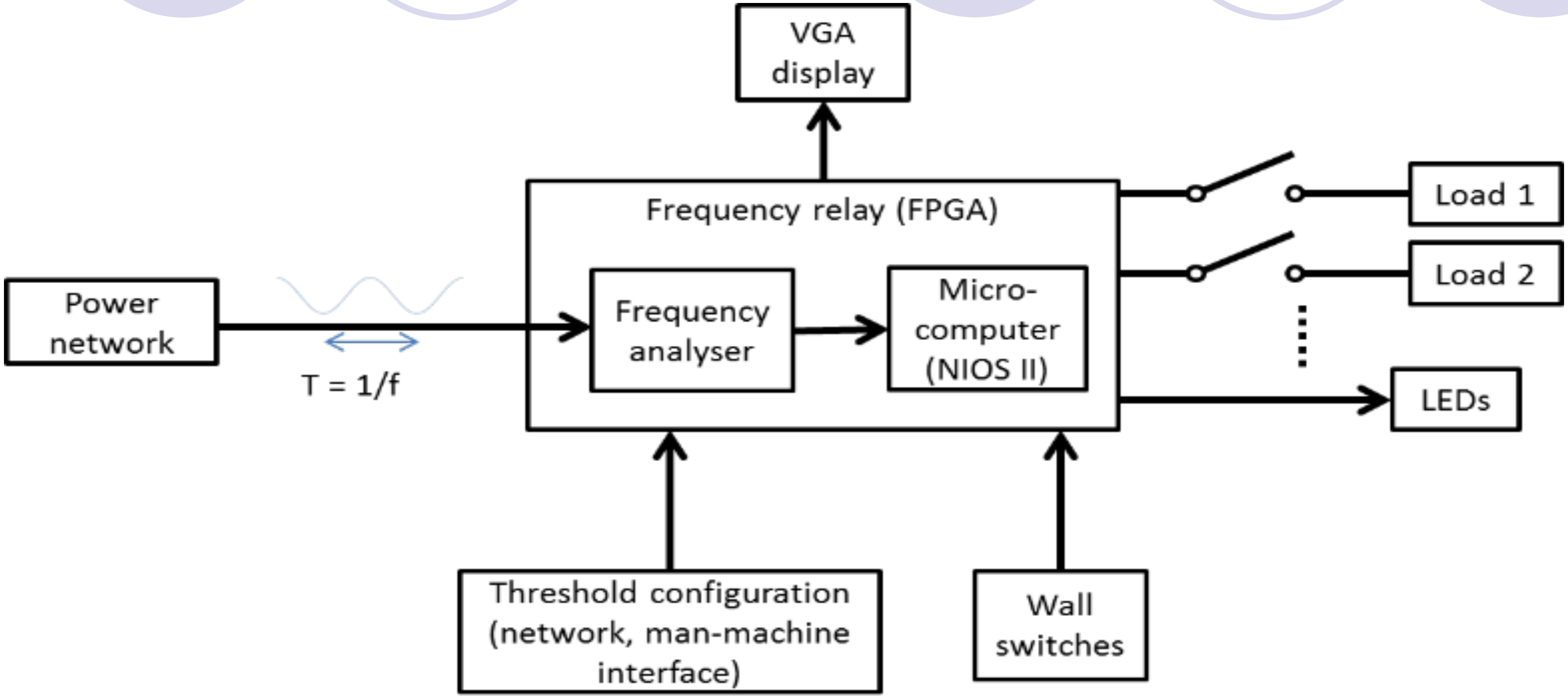


Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- Natural/manual partitioning of functions of frequency relay on
- Hardware implementation (because of speed requirements) – yellow blocks
- Software implementation (because of complexity and combination of control actions) – red blocks



Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System



- Besides main physical input – powers systems signal there are additional inputs and outputs to the FR functionality (UI interface, actuation signals to the physical network, auxiliary UI signals on development board; Internet connectivity will be omitted)

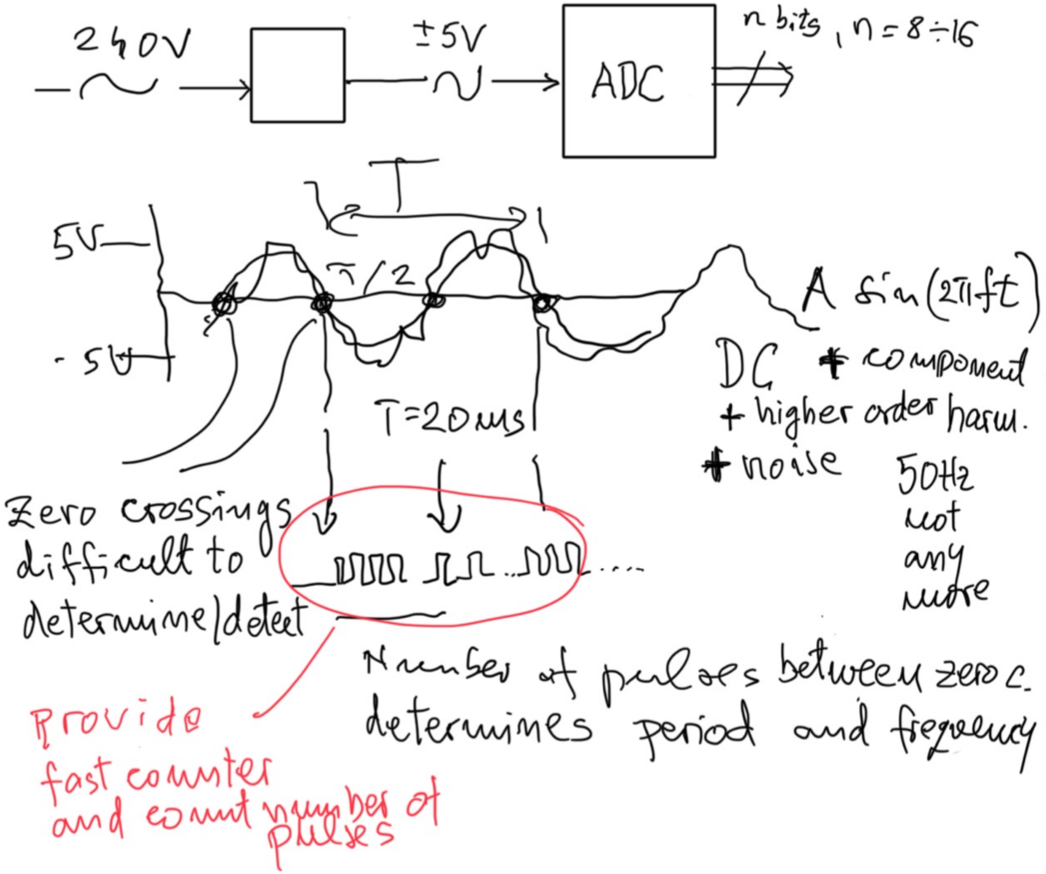
Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- Power signal far from ideal – not looking like sinusoid
- Third, fifth and seventh order harmonics and DC component present + noise
- Hard to measure the frequency from such signal
- We invented multiple methods that allow measurements in very short time (one power signal cycle T=20ms or even half a cycle T/2=10ms)
- Instead of zero-crossings, we find the times of the peaks of the power signal (minimums and maximums)

Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- Time between peaks measured using start/stop of the fast clock of a digital counter;
- The count between two peaks calculated using hardware implemented signal processing → data-dominated part of the system
- The rest (F, ROC) done in software in the microprocessor; all control decisions done in software →control-dominated part of the system
- Other parts of the system include
  - man-machine interface (setting thresholds of tripping the circuits using a keyboard and visualisation on VGA display
  - connection to Internet (not done in the assignment)

Internet-enabled Frequency Relay - - Explanation of Fundamentals



DC + component + higher order harm. + noise

Zero crossings difficult to determine/detect

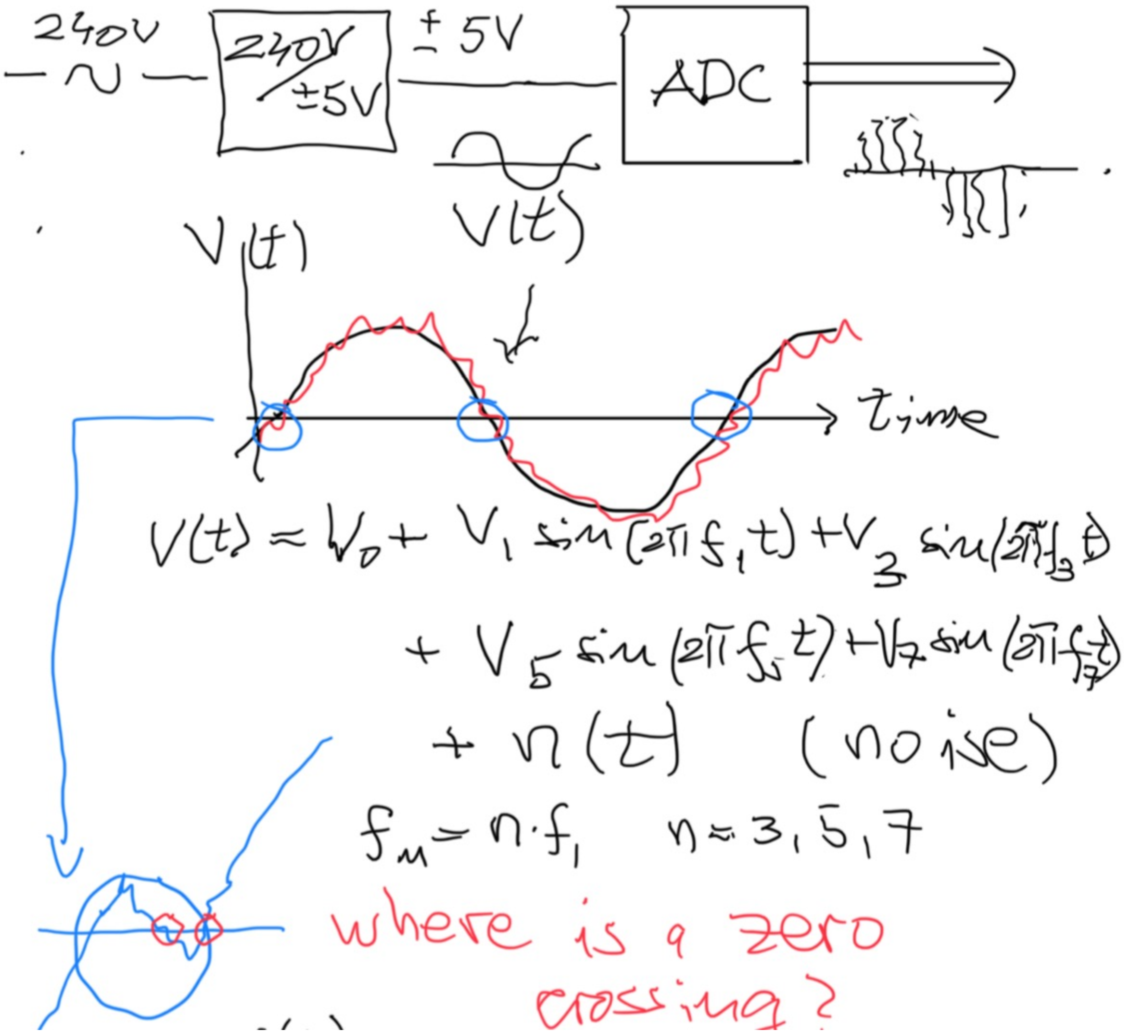
Number of pulses between zero c. determines period and frequency

Provide fast counter and count number of pulses

Internet-enabled Frequency Relay -
- Explanation of Fundamentals

- Basics of sampling and signal processing
  Dealing with higher level harmonics above 50 Hz (3*50, 5*50, 7*50 etc) which can different phase (not in phase with 50 Hz signal)
- DC component and noise (become issue with all electronic devices connected to power network)
- Selection of sampling frequency/period
  Signal processing requires sophisticated algorithms

Internet-enabled Frequency Relay - - Explanation of Fundamentals



$$V(t) = V_0 + V_1 \sin(2\pi f_1 t) + V_3 \sin(2\pi f_3 t)$$

$$+ V_5 \sin(2\pi f_5 t) + V_7 \sin(2\pi f_7 t)$$

$$+ n(t) \quad \text{(noise)}$$

$$f_n = n \cdot f_1 \qquad n = 3, 5, 7$$

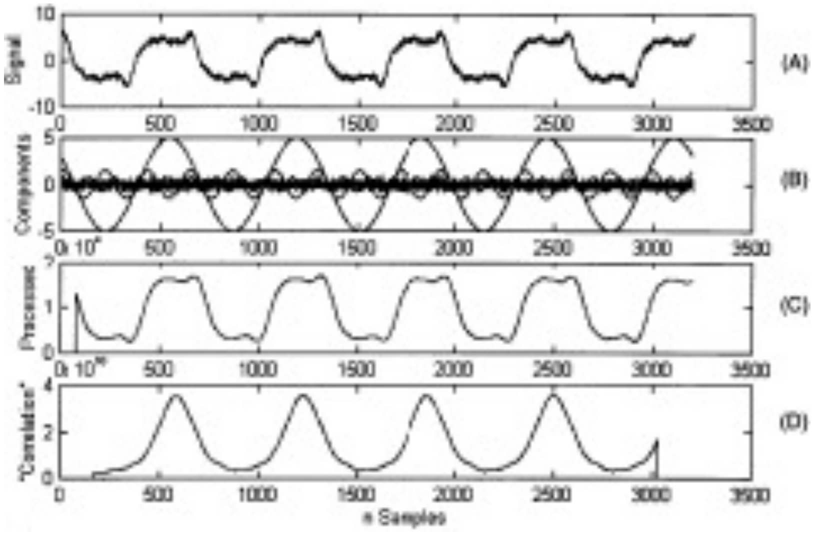
where is a zero crossing?

Internet-enabled Frequency Relay - - Explanation of Fundamentals

- How to determine zero crossings precisely?
  We invented a method to determine precise points in time where the waveform passes extreme points (maximum and minimums) using three signal processing algorithms in series
  - Averaging samples of the original waveform using moving window approach, effectively "smoothing" signal and filtering significant amount of noise
  - Calculating "reference" function which a kind of auto-correlation functions on filtered signal that further cancels noise almost completely and emphasises minimums and maximums of the reference function by changing sign of the gradient of functions precisely at the points of passing maximums and minimums
  - Detecting the change the sign of gradient and generating signal for beginning/end of counting

Explanation of function in MatLab Simulation Experiment

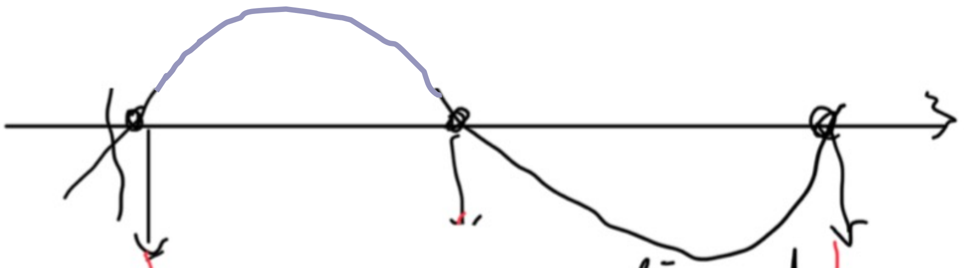
- Input signal
- Components
- Averaged
- Autocorrelation (ref. point function) with the peaks distanced by the period of the input signal



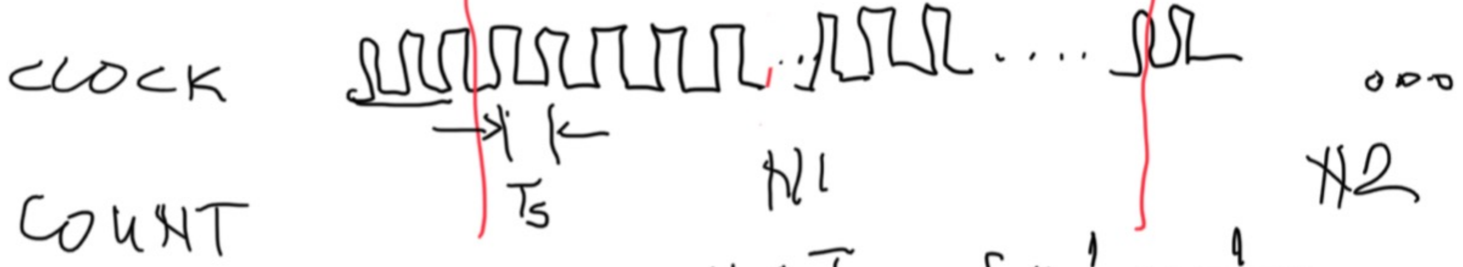
- Input signal modelled with

$$v(t)=0.3+5\sin(w_0t+2.5)+1.5\sin(3w_0t+1.3)+0.75\sin(5w_0t+1.0)+0.375\sin(7w_0t+0.6)+0.1875\sin(9w_0t+0.3)+\text{rand}(\text{SNR}^\circ 21\text{ dB})$$

Internet-enabled Frequency Relay -
- Explanation of Fundamentals

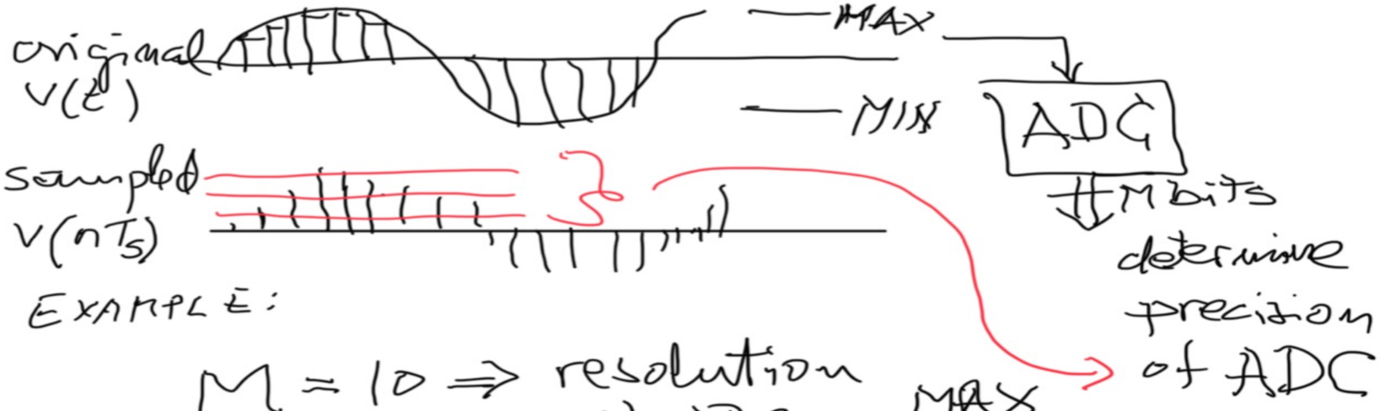


At each zero crossing - activate counting of a "fast"/precise clock



$$T1 = N_1 * T_s \qquad f_1 = \frac{1}{T_1} = \frac{1}{N_1 T_s}$$

Internet-enabled Frequency Relay - - Explanation of Fundamentals

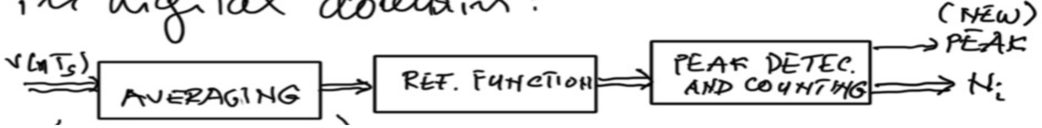


EXAMPLE:

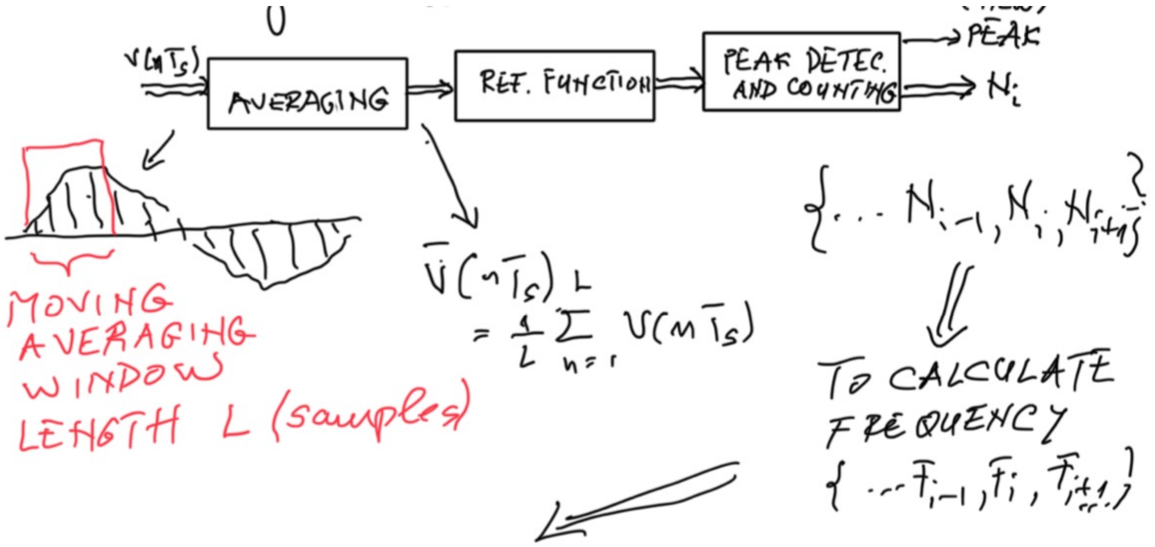
$$M = 10 \Rightarrow \text{resolution of ADC} = \frac{MAX}{2^{10}} = \frac{MAX}{1024}$$

$v(nT_s)$ are samples/series obtained from ADC

ALL further signal processing carried out in digital domain:



Internet-enabled Frequency Relay - - Explanation of Fundamentals



$$\bar{V}(nT_s)_L = \frac{1}{L}\sum_{n=1} V(nT_s)$$

NEW F_i CALCULATED AFTER NEW PEAK DETECTED AND AFTER N_i SAMPLES PROCESSED

NEW ROC (RATE OF CHANGE)

$$ROC = \frac{F_i - F_{i-1}}{\Delta t}$$

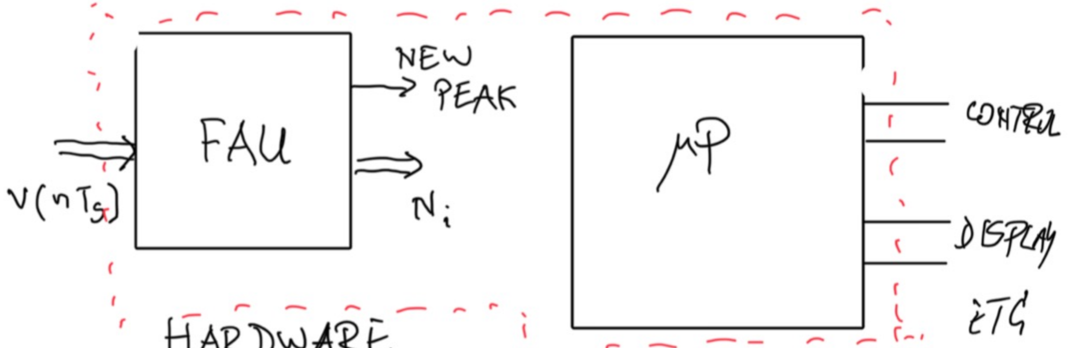
TIME BETWEEN TWO CONSEQ. PEAKS

Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- Implementation is based on hardware/software co-design (partitioning)
- Software-only solution not feasible (hard to guarantee real-time constraints);
- Partitioning done by the designer at the boundary of data-dominated and control-dominated part
- Control dominated part driven by interrupts from Frequency Measurement Unit (FMU), keyboard (Internet), DMA controller in the microcomputer configuration
- Implementation in FPGA (both HW and SW part)

Internet-enabled Frequency Relay - Implementation – HW/SW Co-Design

Input from ADC



Switches, TRIACS

VGA, LEDs

HARDWARE IMPLEMENTED (FPGA)

SOFTWARE (TASKS, ISRs) AROUND FreeRTOS

$f_s = \frac{1}{T_s}$ sampling frequency

Higher better

In your assignment "only" 16 kHz

- EARLY DESIGN WAS ON MC68HC11 (SEPARATE CHIP)
- NEW DESIGN ON NiOS II (IN FPGA)

Single sequential program with ISR

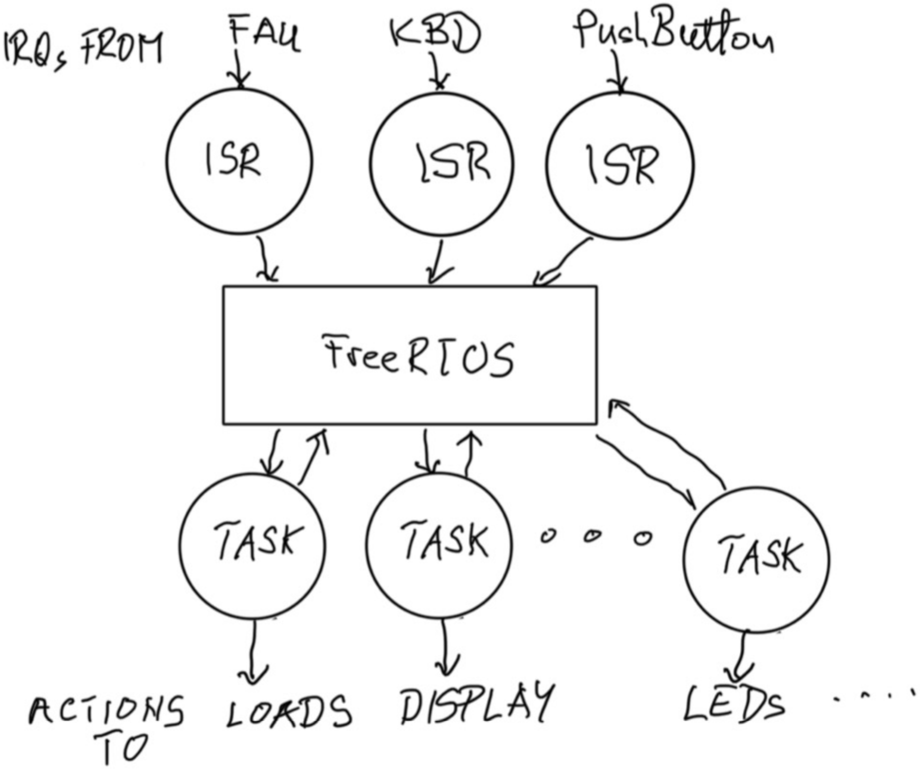
Multi-tasking implementation - FreeRTOS

Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- Test-bed created by using “emulation” of power signal with all “nasty” features
- Power signal can be replicated by reading samples from FPGA on-chip memory – otherwise we could wait for years a real event to happen ☺ ☹
- SW part as a solution using multiple tasks and ISRs in FreeRTOS setting

Internet-enabled Frequency Relay - Software Implementation

Inputs (Interrupt requests, #samples, key's codes)



Outputs (turn-on/off loads using TRIACs,…)

Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

- No mentioning of Internet !!
- How can we incorporate Internet access to the LCFR?
  - Simplest way is to run a small web-server on the Microprocessor platform and enable remote access for various kinds of services (e.g. a task on FreeRTOS)
  - However, for this we need physical access to Internet (via Ethernet and local router; Ethernet port exists on DE2-115 board))
  - The story now goes on….
  - This was an approach we made before notion of Internet of Things (IoT) was introduced

Frequency Relay Simulator

Frequency Relay - Simulated case
No development kit – using FreeRTOS simulator
Not done in 2024

Software only

Details of the theory behind the FR are in

Z Salcic, R Mikhael, 2000. A new method for instantaneous power system frequency measurement using reference points detection, Electric Power Systems Research, Volume 55, Issue 2, 1 August 2000, Pages 97-102, https://doi.org/10.1016/S0378-7796(99)00102-9

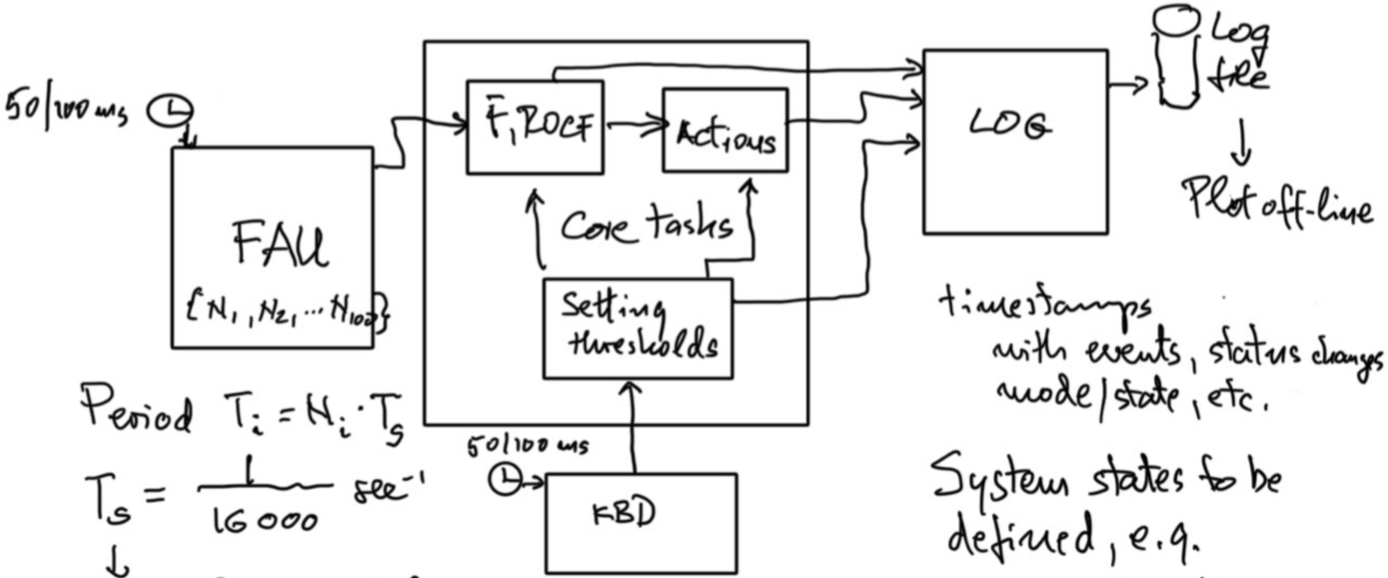
Frequency Relay - Simulated case
No development kit – using FreeRTOS simulator

- Conceptual design, no ISRs
- Core tasks for calculating F and ROCF, undertaking actions (load management), changing state, parameters
- Environment simulation tasks:
  - FAU and maintaining system time with the resolution of cca 20 ms,
  - getting input from a keyboard for change of thresholds TF and TROCF as well as of change of system state and
  - collecting a log of events in the system (with system time timestamp for each event, change)
- System time is calculated and maintained as a global variable by FAU task and can be read by all other tasks

Frequency Relay - Simulated LCFR No development kit – using FreeRTOS simulator

- FAU task simulates FAU unit operation by generating a series Ni of numbers that represent a number of samples/clock cycles of a sampling clock of 16 KHz between two peaks of the power signal; samples are provided for 2 sec of power system operation; FAU is a periodic task
- KBD task receives inputs from a keyboard via stdio function (e.g. thresholds, change of LCFR state/mode) and communicates typed characters to core tasks; KBD is a periodic task with period much faster than the time of typing single character
- LOG task receives information on events/actions and state of the loads to a Log that can be used for plotting ....

Frequency Relay - Simulated LCFR
No development kit – using FreeRTOS simulator



Period $T_i = N_i \cdot T_s$

$T_s = \frac{1}{16000}$ sec^{-1}

↓

Sampling period

$f_s = 1/T_s$ =16 KHz

Time starts with 0 and progresses as discrete time series

$t_k = t_{prev} + T_k$ $k=2,\ldots$

$t_1 = T_1$, $t_{prev} = t_1$

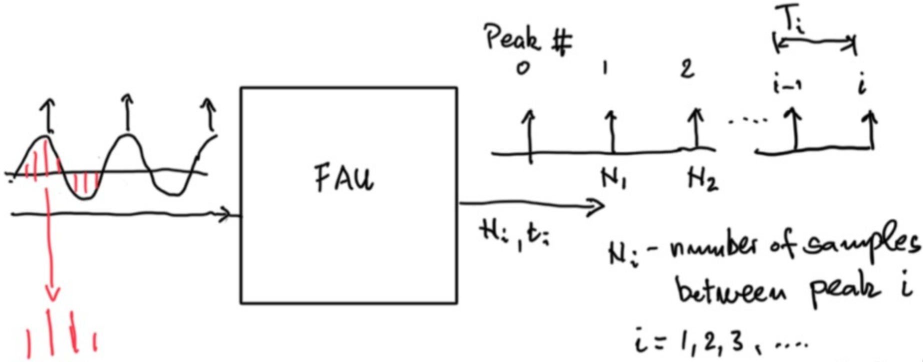
Use roughly 50 ms or 100 ms of simulator time to simulate ~20 ms of LCFR time

timestamps
with events, status changes
mode/state, etc.

System states to be defined, e.g.

* Initialisation
* Normal operation
* Maintenance

Frequency Relay - Simulated LCFR
No development kit – using FreeRTOS simulator



N_i - number of samples counted between peak i and $i-1$

$i = 1, 2, 3, \ldots$

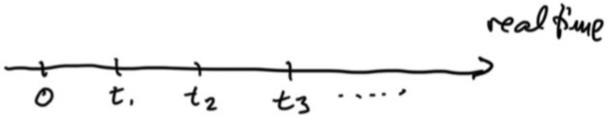
$T_i = N_i \cdot T_s$ ⇒ instanteneous frequency $f_i = \frac{1}{T_i}$

$T_s = \frac{1}{16000}$ $\sec^{-1}$

$f_s = \frac{1}{T_s} = 16000\,Hz$
$= 16\,KHz$

System time at peak i is
$t_i = T_1 + T_2 + \ldots + T_i$
and it is real time

$t_1 = T_1$
$t_2 = T_1 + T_2$



We simulate FAU by generating N_i, t_i where t_i is discrete system time within FAU task
Each N_i has corresponding f_i

$$\begin{array}{ccccc} \{N_1, & N_2, & \ldots, & N_{100}\} \\ \updownarrow & \updownarrow & & \updownarrow \\ f_1 & f_2 & & f_{100} \end{array}$$

Internet-enabled Frequency Relay - Example of a Heterogeneous Embedded System

The assignment done in groups/teams of two students

Happy Designing ☺