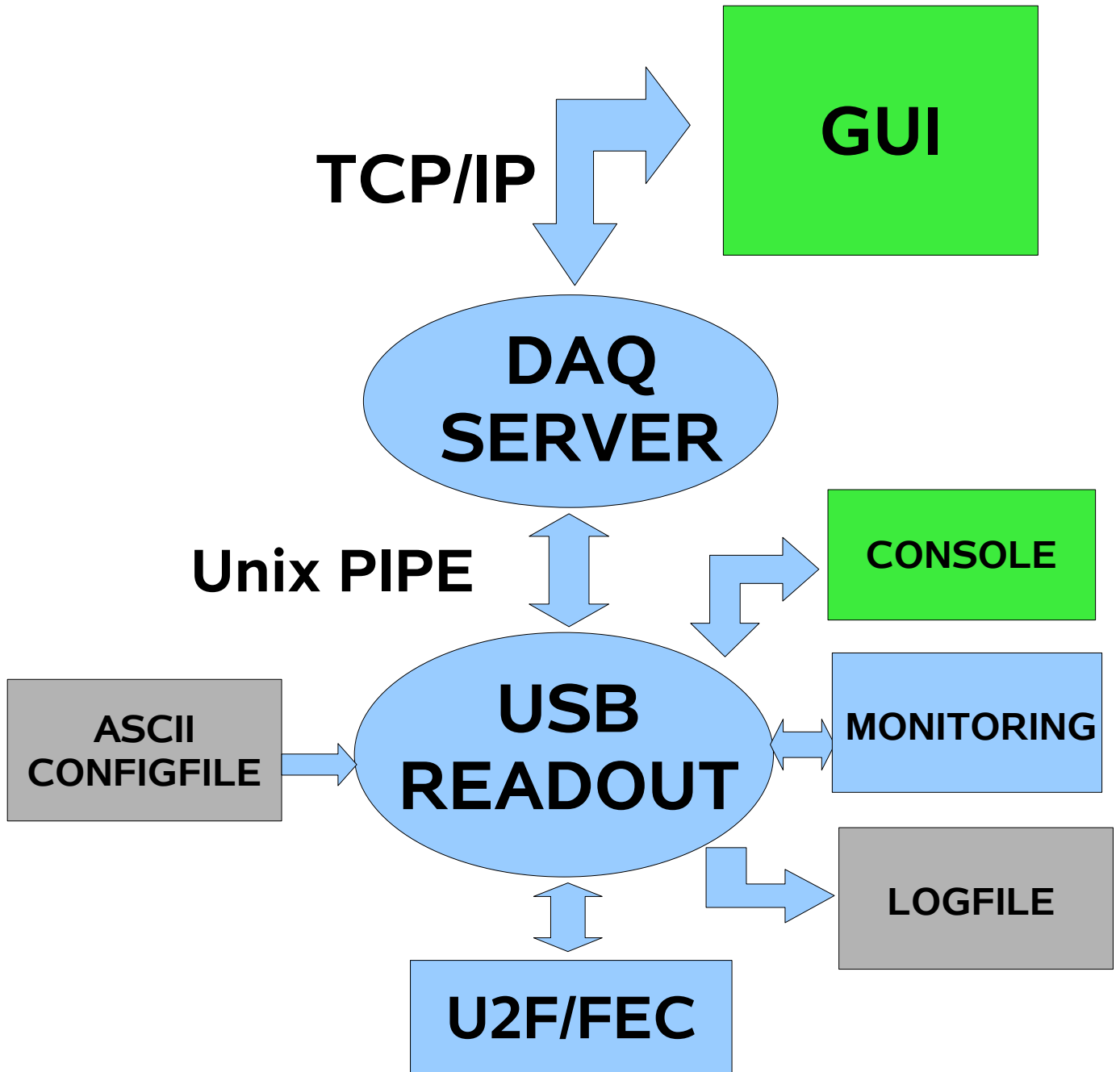


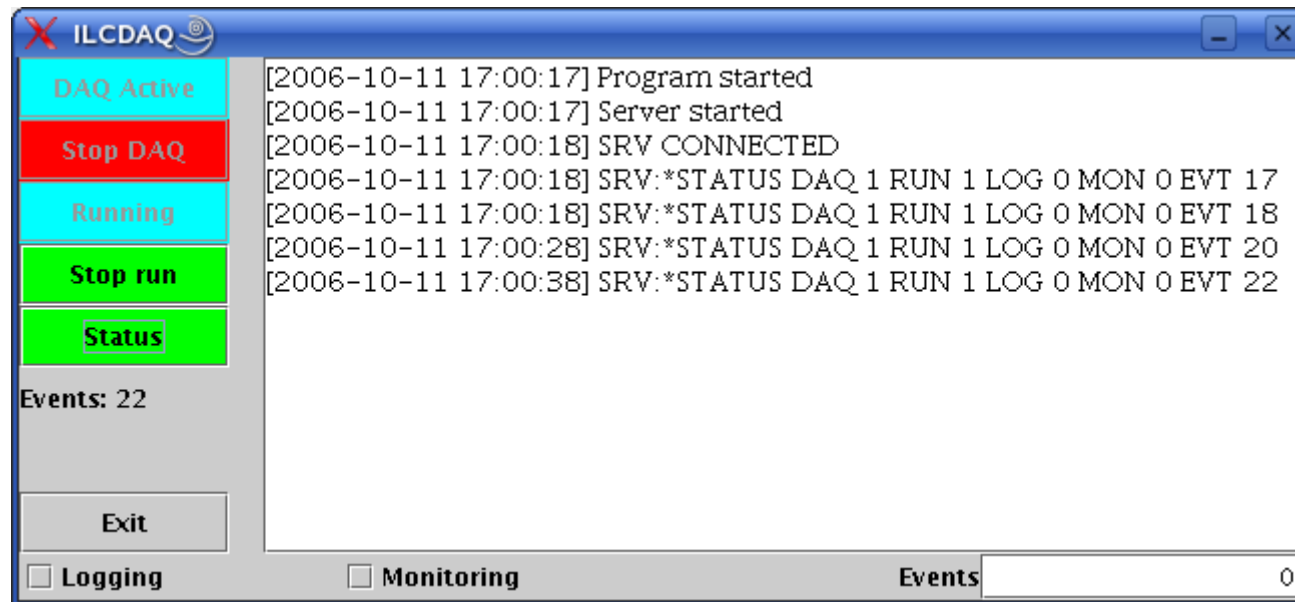
# A simple Desktop DAQ for U2F readout

## **Status and plans**

Ulf jörnmark  
Physics Dept.  
Lund



# Graphical User Interface (Java)



## ***DAQ SERVER and USB READOUT***

### **COMMUNICATION METHODS**

Between GUI and DAQ server: Network (TCP/IP)

Between DAQ Server and USB Readout: Unix signal and named pipe

### **Simple commands exchanged:**

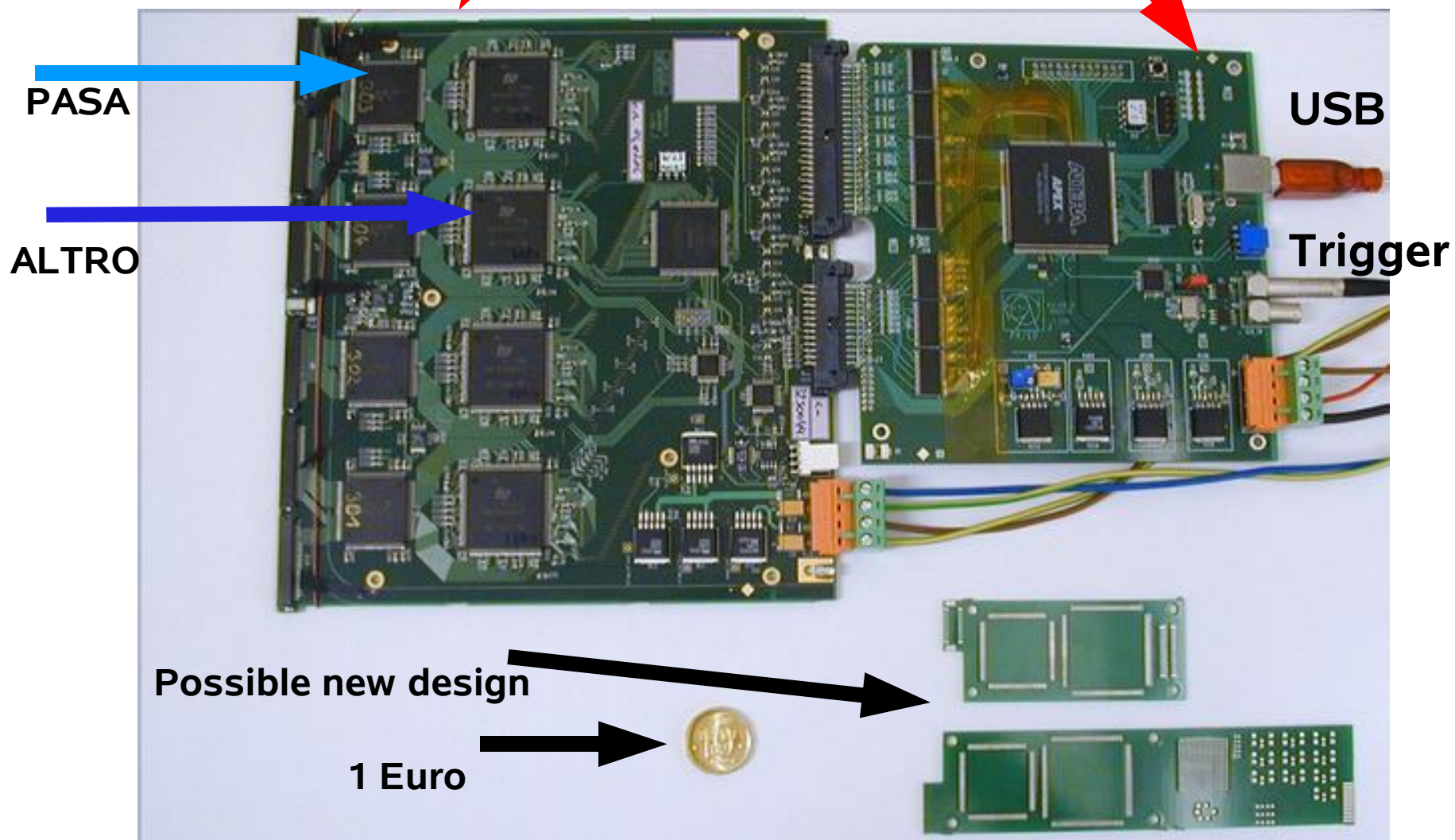
- \*START: Start system and initialize front end, e.g. power on FECs
- \*SOR : Start Of Run
- \*EOR: End of Run
- \*STOP: Stop system, e.g. power off FECs
- \*STATUS: request/send status information

### **Console:**

Simple menu with actions

# ALICE FEC & U2F cards

Current setup: 1 FEC + 1 U2F



## ***U2F READOUT***

- Trigger to U2F card starts readout of FEC
- Two readout modes of U2F card through USB:
  - polled: ALTRO data is read channel by channel – SLOW
  - pushed: Data driven – FAST
- Code: Example C code from the BoNus experiment

## ***U2F READOUT***

In *poll* mode must the computer react on the trigger.  
Large overhead in the USB calls for each channel to be read makes it slow.

In *push* mode is all data sent from U2F as soon as it is available (in  $\leq 1024$  byte bursts) =>  
Computer do not need a trigger.  
Can have several events in the same data buffer.  
Must scan for the end of event marker to separate the events.  
Small overhead in USB call makes it fast.  
BUSY handling?

## ***DATA LOGGING***

Written to binary file as received (raw data format)

## ***MONITORING***

Pending. Plan to use ROOT.

## ***INTEGRATION INTO GENERAL DAQ?***

Pushed mode a problem?