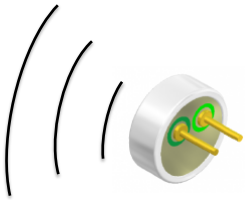
ETH Course 402-0248-00L: Electronics for Physicists II (Digital)

- 1: Setup uC tools, introduction
- 2: Solder SMD Arduino Nano board
- 3: Build application around ATmega328P
- 4: Design your own PCB schematic
- 5: Place and route your PCB
- 6: Start logic design with FPGAs

Exercise 3: "Sound volume robot"

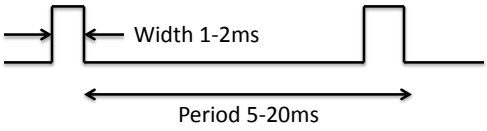
- measures sound volume and moves arm to indicate loudness
- microphone -> preamp -> ADC -> uC -> PWM output

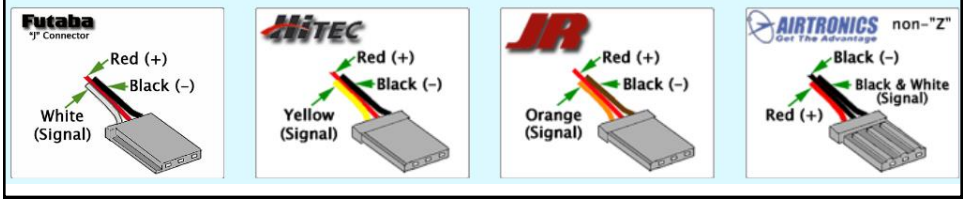


(debugging, programming)

"RC" servos (Radio-Control Servo-Motors)

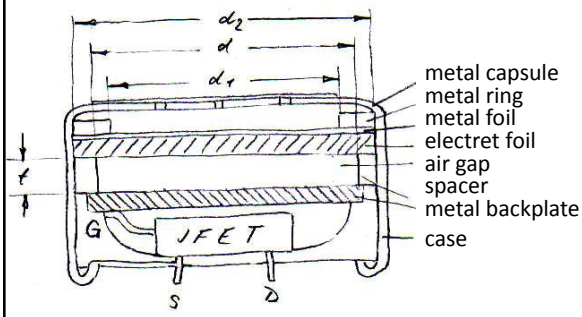
- Position controlled – Servo has internal position measurement and controller
- Rotation angle 120 degrees
- Pulse width from 1-2ms sets desired position
- Pulses must be sent at frequency 50-200Hz
- Pulse height >2V

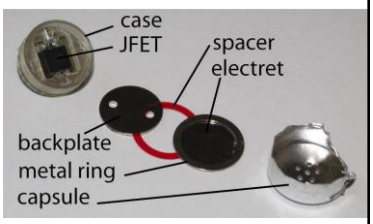




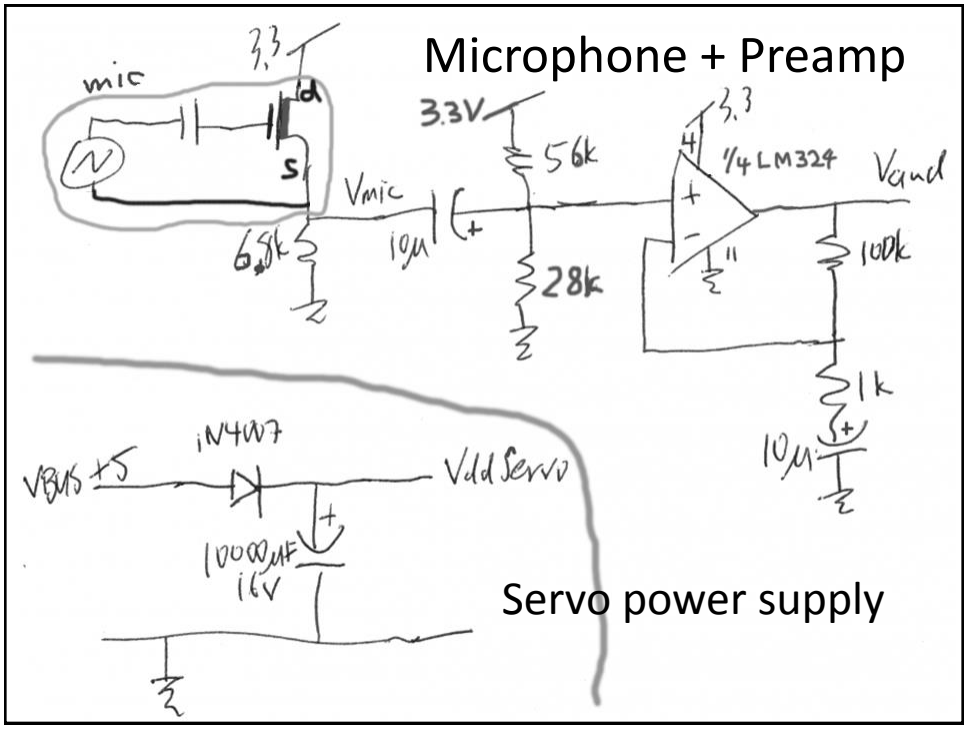
Electret Microphone

- Cheap (< 1$)
- Electret material, no polarization voltage is required
- Low-noise JFET buffer
- Metal foil is connected to source of the JFET through metal capsule



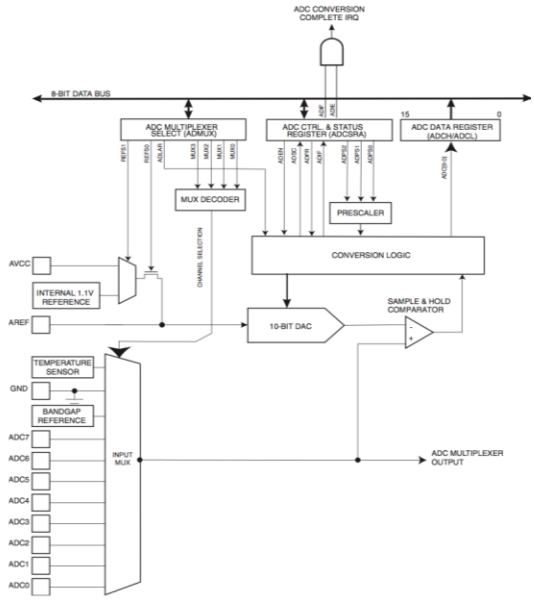


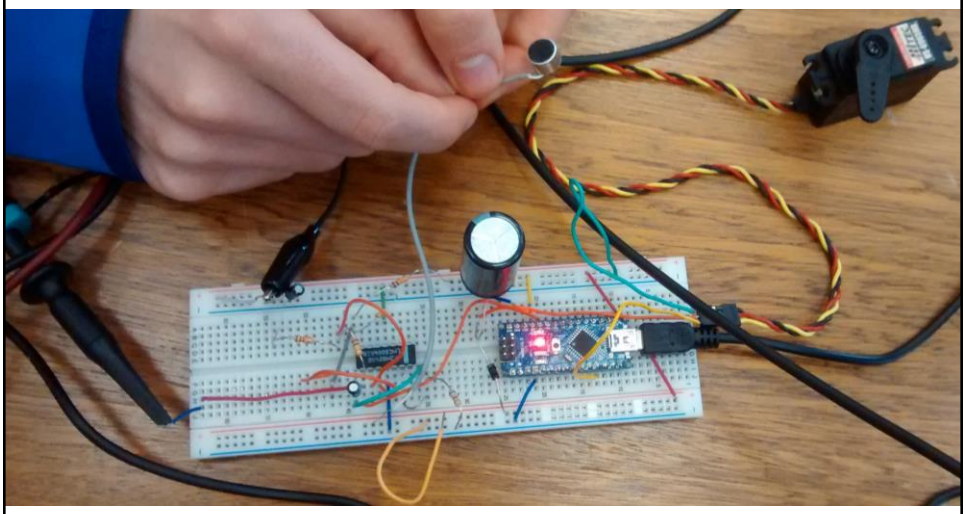
4



ATmega328P Analog to Digital converter

- 10-bit Successive approximation register (SAR) type
- 8 multiplexed single-ended input channels
- Internal Temp sensor
- Max combined sample rate 79.6ks/s
- Interrupt on End of Conversion.
- Triggered by:
  - External Interrupt Request 0
  - Timer 0
  - Timer 1
  - Analog Comparator

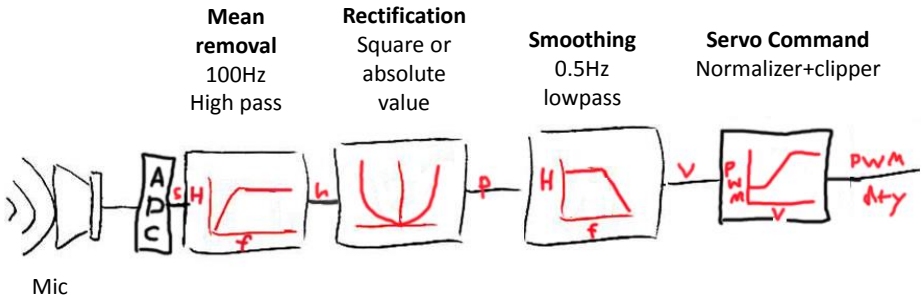




- Fixed-point digital signal processing pipeline
- Using timer interrupts for regular ADC sampling intervals

Signal processing pipeline

produces servo position corresponding to average sound volume

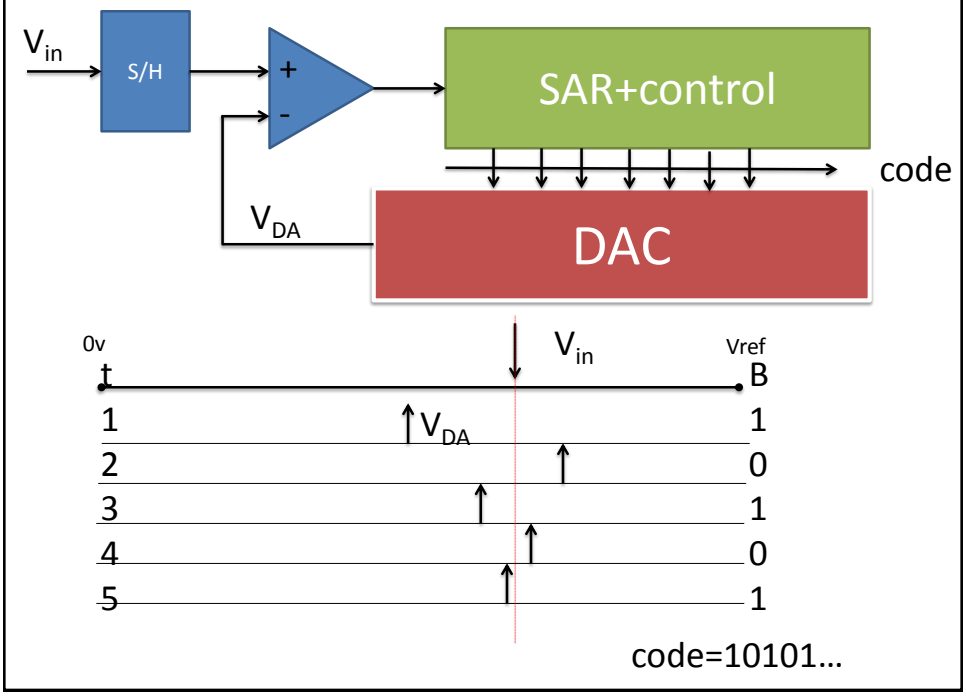


Some more about ADCs

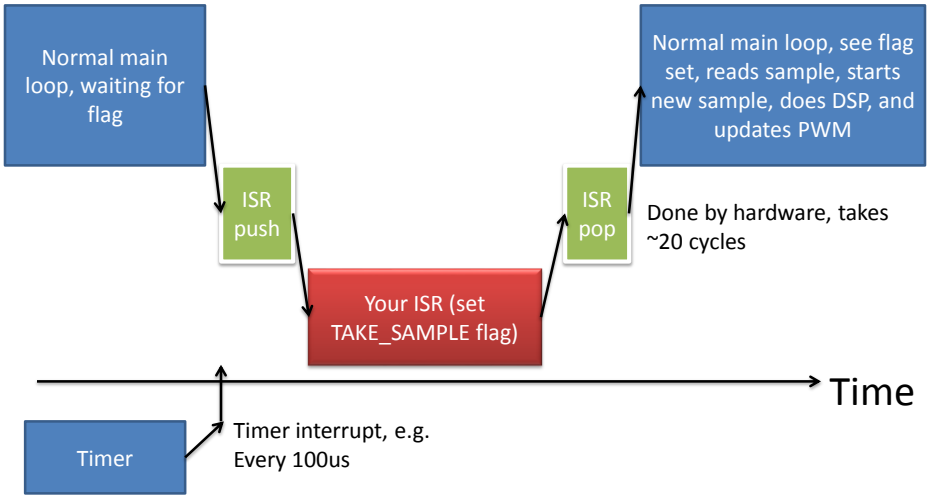
High resolution Low speed and power	Medium resolution Medium power	Low resolution but fast and hot
Single slope (imprecise)	SAR (good tradeoffs, most uC)	Flash (video rate, oscilloscopes)
Dual slope (precise but very slow)	Algorithmic ($\Sigma\Delta$)	2-step

ADC specifications

INL	Integral nonlinearity	Max absolute sample deviation in bits
DNL	Differential nonlinearity	Max possible step size variation in bits
Sample rate		
Latency	In samples	How long in samples it takes for a conversion (can be >>1 for pipelined converter)
Reference voltage	Volts	Minimum resolution



Using timer overflow interrupt for regular ADC sampling intervals in an Interrupt Service Routine (ISR)



Initialize ADC and Timer (and make the timer run) in setup()

Timer Counter (TC) 2 setup

- Download MsTimer2.zip (from lab3) and unzip in your Arduino/libraries folder.
- Add #include <MsTimer2.h> at the beginning.
- Setup(): Add the following lines:

  MsTimer2::set(time in us,t2_ovf);

  MsTimer2::start();
- From now, for each Timer2 overflows, t2_ovf() will be executed. You need to declare and write code for t2_ovf() function.

ISR

```
void t2_ovf(void) {
  // Increment the counter, which is also
  used to determine servo updates
  tc_tick++;

  // set a flag to tell main loop to take a
  sample
  takeSampleNow = TRUE;

  // Toggle a GPIO pin (this pin is used as a
  regular GPIO pin).
  digitalWrite(13,!digitalRead(13)); //
  debug, should toggle at desired sample rate
}
```

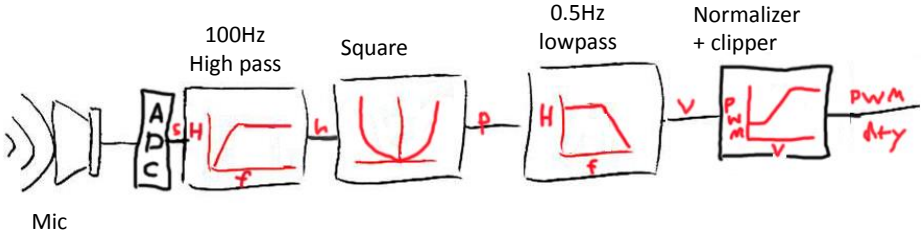
MsTimer2.h

```
#include <avr/interrupt.h>

namespace MsTimer2 {
    extern unsigned long msecs;
    extern void (*func)();
    extern volatile unsigned long count;
    extern volatile char overflowing;
    extern volatile unsigned int tcnt2;

    void set(unsigned long ms, void (*f)());
    void start();
    void stop();
    void _overflow();
}

#endif
```

MsTimer2.cpp

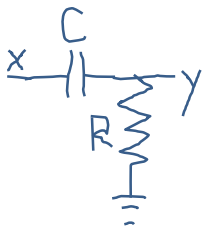
```
#include <MsTimer2.h>

unsigned long MsTimer2::msecs;
void (*MsTimer2::func)();
volatile unsigned long MsTimer2::count;
volatile char MsTimer2::overflowing;
volatile unsigned int MsTimer2::tcnt2;

void MsTimer2::set(unsigned long ms, void (*f)()) {
    float prescaler = 0.0;

#if defined (__AVR_ATmega168__) || defined (__AVR_ATmega48__) || defined (__AVR_ATmega88__) || defined (__AVR_ATmega328P__)
    TIMSK2 &= ~(1<<TOIE2);
    TCCR2A &= ~((1<<WGM21) | (1<<WGM20));
    TCCR2B &= ~(1<<WGM22);
    ASSR &= ~(1<<AS2);
    TIMSK2 &= ~(1<<OCIE2A);

    if ((F_CPU >= 1000000UL) && (F_CPU <= 16000000UL)) {    // prescaler set to 64
        TCCR2B |= (1<<CS22);
        TCCR2B &= ~((1<<CS21) | (1<<CS20));
        prescaler = 64.0;
    } else if (F_CPU < 1000000UL) { // prescaler set to 8
        TCCR2B |= (1<<CS21);
        TCCR2B &= ~((1<<CS22) | (1<<CS20));
        prescaler = 8.0;
    } else { // F_CPU > 16Mhz, prescaler set to 128
        TCCR2B |= ((1<<CS22) | (1<<CS20));
        TCCR2B &= ~(1<<CS21);
        prescaler = 128.0;
    }
```

Fixed point signal processing pipeline



We need a digital low & high pass filters, like an RC or CR filter

A simple IIR high pass filter (discrete time)



$$\frac{y}{R} = C\left(\dot{x} - \dot{y}\right)$$

$$RC\dot{y} + y = RC\dot{x}$$

$$\tau\dot{y} + y = \tau\dot{x}$$

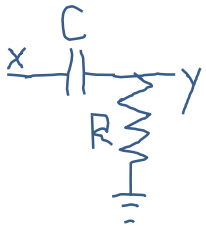
$$\tau\left(\frac{y_{t+\delta t} - y_t}{\delta t}\right) + y_t = \tau\left(\frac{x_{t+\delta t} - x_t}{\delta t}\right)$$

$$\alpha = \frac{\delta t}{\tau}$$

$$y_{t+\delta t} = y_t - \alpha y_t + x_{t+\delta t} - x_t$$

$$= (1-\alpha)y_t + x_{t+\delta t} - x_t$$

A simple IIR high pass digital filter
(fixed point, using binary shift operations)



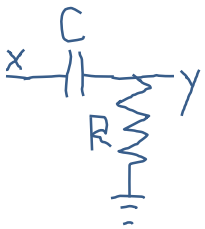
$$y_{t+\delta t} = (1-\alpha)y_t + x_{t+\delta t} - x_t$$

If $\alpha = \frac{1}{2^n}$, then

$$(1-\alpha)y_t = \frac{2^n - 1}{2^n} y_t = \left[\left(y_t \ll n \right) - y_t \right] \gg n$$

$$y_{t+\delta t} = \left[\left(y_t \ll n \right) - y_t \right] \gg n + \left(x_{t+\delta t} - x_t \right)$$

What is the time constant?



$\alpha = \frac{\delta t}{\tau}$

Suppose $\delta t = 100$us (10kHz sample rate)

and $\alpha = 1/256$ (n=8).

Then

$\tau = 100$us x 256=25.6ms

Corner frequency $f_{3dB} = \frac{1}{2\pi\tau} = 6.2Hz$

To filter with n times longer time constant, you can skip n samples

DSP code sample

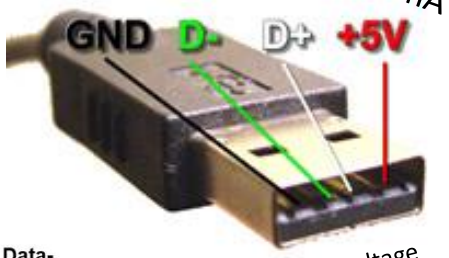
```
void device_task(void) {

if (takeSampleNow) {  // flag set in timer ISR
takeSampleNow=FALSE;
// signal processing
int adcval = analogRead(apin); // 0-1023=5V

if (initialized)
        audMean = ((adcval-audMean)>>NTAU1)+audMean; // TODO mix old and new value
else
        audMean = adcval; // init filter with first reading

// only update meanSq at TAU2 interval, so to produce effective time constant that
is TAU2 times tau of audMean filtering
if(dspCounter--==0){
    dspCounter=TAU2;
    long diff = adcval - audMean; // signed diff of sample from mean
    long sq = diff * diff; // square diff
    if (initialized)
        meanSq = ((sq-meanSq)>>NTAU1)+meanSq; // low pass square diff
    else
        meanSq = sq;
}
}
}
```

USB – Universal Serial Bus

- Physical layer
- User perspective (coder)
- Under the hood
  - Device side
  - Host side
- Achieving high performance

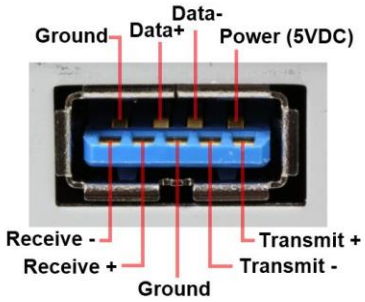
USB Physical layer

- Up to USB 2.0 – full (12Mbps) and high (480Mbps) speed
- USB 3.0 super speed (5Ggbs)



Up to 200mA

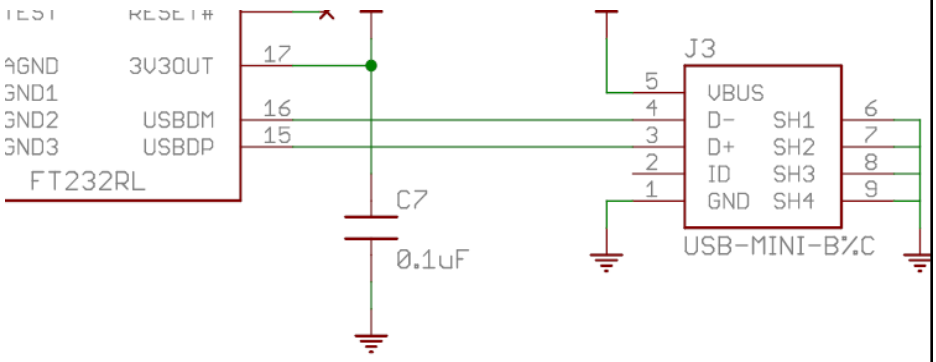
Low voltage differential signal



USB definitions

- IN means towards the host (the PC)
- OUT means towards the device (uC)

USB on PCB



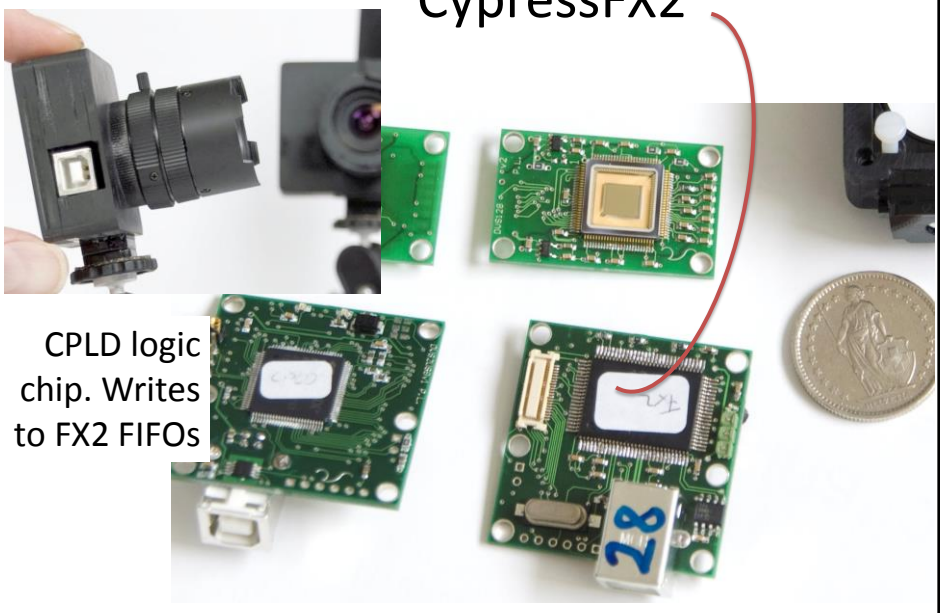
USB performance

- USB full speed (12Mbps): about 1MBps
- USB high speed (480Mbps): about 40MBps
- USB super speed (5Gbps): ??

ICs for USB

USB full speed	• Many uC. Also FTDI.
USB high speed	• CypressFX2
USB super speed	• CypressFX3

CypressFX2



CPLD logic chip. Writes to FX2 FIFOs

ftdichip.com

- uC UART – USB interface; looks like COM serial port on host side.
- Max speed is only 12Mbaud for the UART port unfortunately

D-PHYS Shop

The Physics Department conducts the D-PHYS shop with a centralized accounting office through which the customers are charged. The D-PHYS shop is open to all members of ETH, but the settlement mode must be cleared in advance with the accounting office. Private purchases in cash are also possible.

Contact
ETH Zurich
D-PHYS-Shop
Department of Physics
HPF B 10.1
Otto-Stern-Weg 1
8093 Zurich
Tel +41 44 633 23 04
dshop@phys.ethz.ch

Opening hours
7.00 - 11.45/12.45 - 16.30

Ordering
Catalog

Staff

Head

Hermann Wüest
ETH Zürich
HPF G 9.4
CH-8093 Zürich
Tel. +41 44 633 25 90
Fax +41 44 633 11 06
wueest@phys.ethz.ch
vCard

Urs Jakob
ETH Zürich
HPF B 10.1
CH-8093 Zürich
Tel. +41 44 633 23 04
Fax +41 44 633 11 43
urs.jakob@phys.ethz.ch
vCard

3 groups of 8-10 people:
1st => 14:50 – 15:20
2nd => 15:20 – 15:50
3rd => 15:50 – 16:20